# A POWER SHARING BASED APPROACH FOR OPTIMIZING PHOTOVOLTAIC GENERATION IN AUTONOMOUS MICROGRIDS

BY

OLAOLUWAPO AJALA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Professor Peter Sauer

# ABSTRACT

In an effort to reduce the electric industry's dependence on fossil fuels, renewable energy resources are being deployed in the power grid. However, the intermittent nature of some popular renewable resources presents the problem of optimizing their use.

This work presents a control strategy for the optimization of available photovoltaic generation in an autonomous microgrid. The deepening penetration of rooftop solar panels motivates our case study of a microgrid with photovoltaic (PV) generators and microturbines.

Building upon centralized generation concepts such as droop control, automatic generation control and area control error, the strategy ensures effective provision of power raise/lower actions to the system based on some locally measured variables, with the ability to turn-off the microturbine in the event of adequate PV generation for such actions.

A 5-bus test system with three PV generators and one microturbine is considered, and preliminary simulation results are presented to demonstrate the effectiveness of this approach

*To my family and friends, for their love and support.*

# ACKNOWLEDGMENTS

I owe a debt of gratitude to my adviser, Professor Peter Sauer, for the invaluable opportunity to study under his guidance and mentorship at the University of Illinois at Urbana-Champaign. His experience-based advice, unique perspective, and effective advising approach has been much appreciated.

I would also like to acknowledge the Grainger Foundation for providing funding and encouragement to complete this work. Also, thank you to all my friends and family for your love, support and encouragement.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| PV | Photovoltaic |
| MPP | Maximum Power Point |
| MPPT | Maximum Power Point Tracking |
| PLL | Phase Lock Loop |
| ESS | Energy Storage System |
| PCC | Point of Common Coupling |
| DER | Distributed Energy Resource |
| DG | Distributed Generation |
| ACE | Area Control Error |
| ACS | Area Control Strategy |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

The legacy power grid design is based on a centralized generation approach, where generators and load centers are located hundreds or thousands of miles apart and interconnected through transmission and distribution networks. Consequently, an historical solution to the increasing electricity demand has been generation, transmission and distribution infrastructure expansion. Although this approach offers some economies of scale, its implementation is constrained by economic pressures, new environmental policies, and long lead times [3].

On the other hand, distributed generation can ease the pressure on generation and transmission system capacity by supplying some of the load [3]. Distributed generation (DG) is the integration of small scale generation technologies in the distribution network to meet some electricity demand. Common examples of distributed energy resources (DERs) are photovoltaic systems, microturbines, wind turbines and fuel cells. Through distributed generation, the increasing demand for electricity can be addressed while avoiding the aforementioned constraints. However, there are technical limits on the degree to which DERs can be integrated in a power grid [4], and this questions the sustainability of the approach. Also, anti-islanding protection in these units ensures they automatically shut off when a fault is detected [5], and this constrains the reliability benefits that may be gained from these units.

Nonetheless, recent research has shown that a systematic organization and coordinated integration of DG in the form of a microgrid can help mitigate these concerns [6]. The microgrid concept assumes a cluster of loads and microsources that presents itself as a single controllable entity to the main grid, and can operate in grid connected mode or autonomous mode [7]. Through a

microgrid system operator, the output of DERs can be controlled to meet the electricity needs of the main grid, and in the event that a transmission line fault is detected, the microgrid can be isolated and operated independently of the main grid.

In autonomous microgrids, the units are responsible for the voltage control as well as the power sharing and balancing. The role of the power sharing feature is to ensure that all microsources share the load according to their ratings and availability of power from their energy source [3]. However, the common deployment of intermittent renewable energy resources in these systems makes achieving effective and efficient power sharing, while optimizing renewable generation, a difficult and almost impossible task.

This work presents a strategy for overcoming this difficulty, and its effectiveness is demonstrated through a case study on an autonomous microgrid with photovoltaic generators and microturbines.

## 1.2 Previous Work

In an effort to maximize the efficiency of the photovoltaic (PV) system, PV generators are typically operated at their maximum power point. However, as the penetration of these intermittent PV generators increases, the frequency regulation capability (mainly provided by synchronous generators) and the system inertia decrease, which may lead to severer frequency fluctuations under some disturbances [8]. In light of this, numerous strategies have been proposed to reduce the negative impacts of PV generation intermittency. These may be grouped into three categories.

In the first category, an energy storage system (ESS) is installed and operated while keeping the PVs in maximum power point tracking (MPPT) mode. The ESS, which could be a battery, pumped hydro, compressed air energy storage (CAES), supercapacitor or flywheel, is used to smoothen the PV output by storing energy during high generation periods and delivering energy when the MPP drops to low values (references [9] and [10] and the references therein discuss some strategies in this category). The strategies in this category maximize the efficiency of the PV system by ensuring most of the available PV generation is used by the power system. However, a cost analysis performed in [11] shows that this is the least economical approach.

In the second category, the ESS is replaced by a dump load bank while keeping the PVs in MPPT mode. The dump load bank reduces the power injected into the grid during periods of high fluctuations by absorbing the surplus energy generated by the PV system. A dump load typically consists of a resistance and a power flow controller [11]. By dumping part of the generated power to limit power fluctuations, the high cost of installing an ESS is eliminated, but there is a decrease in PV system efficiency and loss of revenue. Nonetheless, [11] shows that this is a more economical approach.

By operating the PV system below MPP during high generation periods, the cost of installing and operating a dump load bank can be avoided. This motivates the next category which involves the use of power curtailment strategies to control the operating point of the PV system. This control can be achieved by modifying the existing control strategy of the power conditioning unit (PCU); therefore, no additional installations are typically required. Reference [11] shows that this approach is more economical than the former.

In light of the cost benefits associated with the use of power curtailment strategies, much work has been reported in this category. Reference [12] describes a fuzzy logic based algorithm for switching between MPPT mode and frequency regulation mode when necessary, but the drawback of these strategy is that the mode-switching action is not adaptive to the power command. In an effort to overcome this limitation, an adaptive method based on the Newton quadratic interpolation (NQI) is proposed in [13], but the adjustability of PV output for frequency regulation is not considered. Consequently, a new frequency regulation strategy is proposed in [8]. In this strategy, the PV output is adjusted based on a frequency droop curve, and an emergency control mode is introduced where the PV output is automatically curtailed when a severe disturbance occurs. Although this strategy is shown to overcome most of the limitations of the aforementioned algorithms, no solution is given to the problem of optimizing PV generation while performing frequency regulation.

## 1.3 Proposed Approach

Although the power curtailment strategies limit power increment fluctuations, power decrement fluctuations cannot be limited by this approach alone. When a sudden drop in MPP occurs, the PV operating point cannot drop to a value above the MPP. Consequently, the rate of decrease of PV system output cannot be limited. This issue motivates the combined use of power curtailment strategies and ESS. Through this approach, power fluctuations can be limited in both directions. When a sudden increase in MPP occurs, the PV system can be controlled to operate below MPP, and when a sudden drop in MPP occurs, the ESS can be controlled to supply necessary energy so the PV output rate of decrease can be limited.

This work presents a power sharing based control strategy for optimizing PV generation while enabling PV systems participate in voltage and frequency regulation within their capabilities. The method is based on an area control strategy where the system is divided into two control areas. An area contains all non-renewable generation, and the other is composed of renewable generation and major loads. The underlying idea for this approach is that by appropriately minimizing the area interchange within generator capabilities, PV generation can be optimized and power quality maintained. By combining a droop control based power curtailment method with an ESS, the strategy limits the rate of change of PV output, and consequently reduces the effect of PV generation intermittency on power system states.

A case study of PV generators and microturbines is presented, and a modified version of the master/slave control strategy [3] is developed for effective real power sharing and bus voltage magnitude control. In addition, an area control strategy for optimizing the use of renewable generation is discussed. Afterwards, we explore the ability to turn off the microturbine when PV generation is sufficient, and finally some simulation results are reported to confirm the validity of this method.

# CHAPTER 2

# MICROGRID MODELLING

A microgrid is an interconnection of loads and distributed energy resources (DERs) within defined electrical and geographical boundaries. A microgrid is viewed as a single controllable entity by the centralized grid and operates in grid-connected or autonomous mode.

In this chapter, mathematical models for a photovoltaic array, a three-phase inverter, a synchronous generator and a power system network are presented, and a reduced order model for the autonomous microgrid is discussed in section 2.5.

## 2.1 Photovoltaic (PV) Array Model

A PV cell is a semiconductor diode that generates an electric current when its p-n junction is exposed to light [1]. A single PV cell produces about 1.5 W. To increase the power output, a multitude of PV cells are interconnected electrically. A group of series-connected PV cells is called a module, a group of series and/or parallel connected PV modules is called a PV panel, and a group of series and/or parallel connected PV panels is called a PV array [14]. The PV cell exhibits a non-linear electrical characteristic which varies with solar irradiance and temperature. As shown in Figure 2.1, a current source in parallel with a single diode, combined with a series and parallel resistance, can be used to model a PV cell/module [15]. However, this model is sufficient for high irradiance levels only, and a two-diode model more appropriately represents the system behavior at low irradiance levels [14]. The single-diode

model of a PV module is mathematically modeled as follows:

$$I_{DC} = I_{PV} - I_D - I_P \tag{2.1}$$

$$I_{DC} = I_{PV} - I_o(e^{\frac{V_{DC}+R_sI_{DC}}{aV_t}} - 1) - \frac{V_{DC} + R_sI_{DC}}{R_p} \tag{2.2}$$

$I_{DC}$ and $V_{DC}$ are the terminal current and voltage of the PV module respectively, $I_{PV}$ is the photo-current generated from incident solar irradiance, $I_o$ is the diode saturation current, $a$ is the diode ideality factor, $V_t = \frac{M_skT}{q}$ is the thermal voltage of a PV module having $M_s$ series connected cells, $k$ is the Boltzmann constant ($1.3806503 \times 10^{-23}$J/K), $T$ is the $p$-$n$ junction temperature, and q is the electron charge ($1.60217646 \times 10^{-19}$C) The five

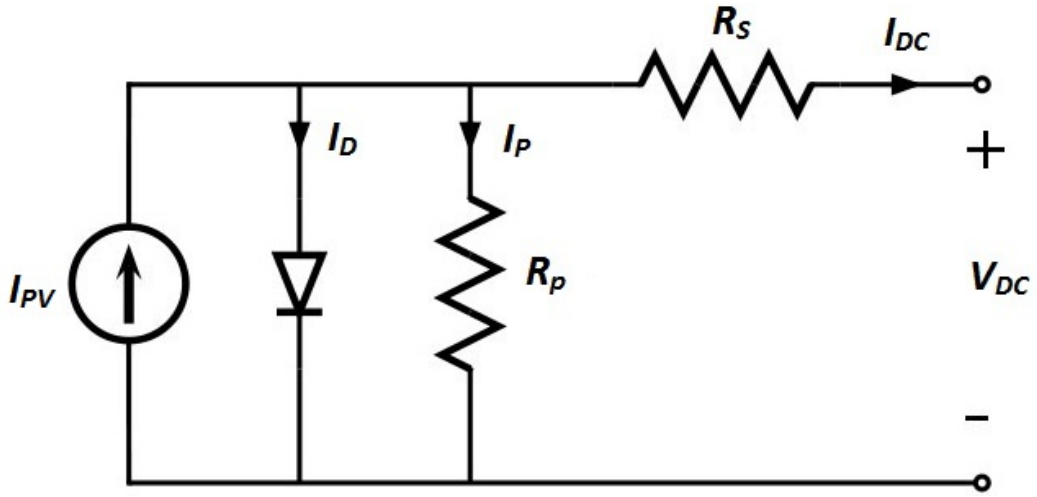


Figure 2.1: Single-diode equivalent circuit model for a photo-voltaic module

circuit parameters, $a$, $I_{PH}$, $I_o$, $R_S$, and $R_P$, are functions of the PV device type. $I_{PH}$ is proportional to the solar irradiance $S$ and temperature $T$, and $I_o$ is a proportional to temperature $T$. $I_{PH}$, $I_o$, $R_S$, and $R_P$ are determined from either experimental measurements or standard testing conditions (STC) operational data provided by the device manufacturer. The value of $a$ is commonly assumed to be between 1.0 and 2.5, depending on the PV device type [14].

$$I_{PV} = [I_{PV,STC} + K_I(T - T_{STC})]\frac{S}{S_{STC}} \tag{2.3}$$

$$I_o = \frac{I_{SC,STC} + K_I(T - T_{STC})}{e^{\left(\frac{V_{OC,STC}+K_V(T-T_{STC})}{aV_T}\right)} - 1} \tag{2.4}$$

STC = standard testing conditions, $S$ = Solar irradiance, $T$ = p-n junction temperature, $S_{STC} = 1000$ W/m$^2$, $T_{STC} = 298.15$ K, $V_{OC,STC}$ is the STC open circuit voltage, $I_{SC,STC}$ is the STC short circuit current, $K_V$ is the open-circuit voltage temperature coefficient, and $K_I$ is the short-circuit current temperature coefficient. At maximum power point (MPP) of a PV module, the following equations hold:

$$I_{DC,mpp} - I_{PV} + I_o(e^{\frac{V_{DC,mpp}+R_s I_{DC,mpp}}{aV_t}} - 1) + \left(\frac{V_{DC,mpp} + R_s I_{DC,mpp}}{R_p}\right) = 0 \tag{2.5}$$

$$\left.\frac{\partial(I_{DC}V_{DC})}{\partial(V_{DC})}\right|_{I_{DC,mpp},V_{DC,mpp}} = 0 \tag{2.6}$$

The result of the partial differential equation above is:

$$I_{DC,mpp} + \frac{I_o(I_{DC,mpp}R_S - V_{DC,mpp})}{aV_t} e^{\left(\frac{V_{DC,mpp}+R_S I_{DC,mpp}}{aV_t}\right)} + \left(\frac{V_{DC,mpp} + I_{DC,mpp}R_S}{R_P}\right) = 0 \tag{2.7}$$

To compute the MPP terminal current and voltage at specified irradiance and temperature values, equations (2.5) and (2.7) may be solved iteratively for the unknowns $I_{DC,mpp}$ and $V_{DC,mpp}$ once $R_S$ and $R_P$ have been evaluated empirically, and $I_{PV}$ and $I_o$ have been computed from equations (2.3) and (2.4) respectively.

If numerous PV modules are electrically interconnected, the net output power is increased. If all the interconnected arrays are of the same type, subject to the same environmental conditions, and if the effects of shading on the PV arrays are neglected, an approximate equivalent circuit may be developed for the PV array system. Unless phenomena specific to a PV array are to be studied, this approximation is sufficient for power system studies [16, 14]. The approximation is shown in Figure 2.2.

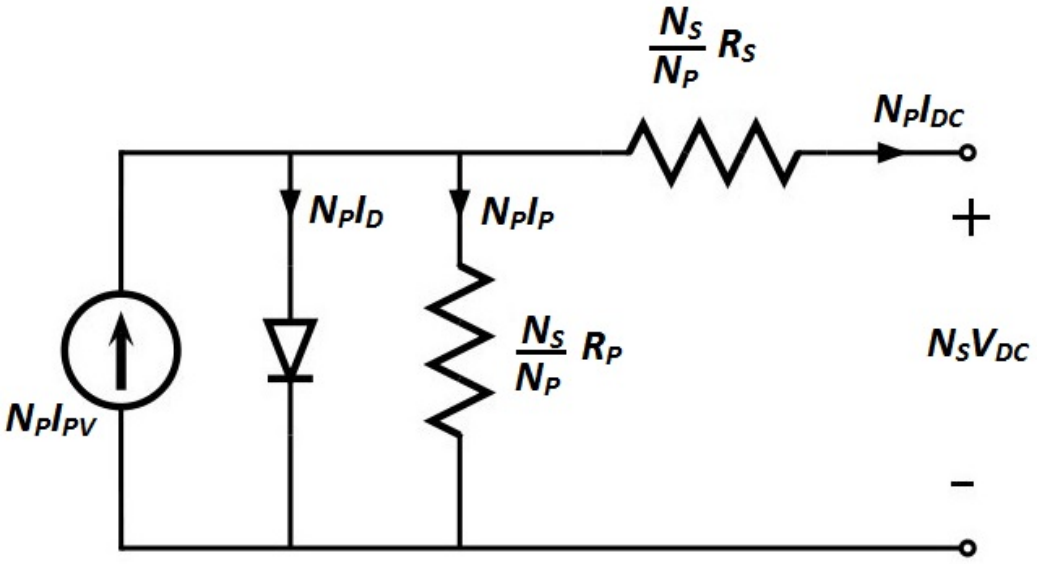


Figure 2.2: Single-diode equivalent circuit model for KC200GT solar array system with $N_S$ series arrays and $N_P$ parallel arrays (thermal voltage $(V_t) = \frac{M_S N_S kT}{q}$) [1]

## 2.2 Three-Phase Inverter Model and Control

An inverter is a DC-AC power converter. It is composed of power electronic switches which may be controlled by a pulse width modulation (PWM) strategy[17]. The PWM strategy compares a high frequency periodic waveform, the carrier signal, with a slow varying waveform known as the modulating signal. When the modulating signal is greater than the carrier signal, a turn-on command is issued to the switches, and when the modulating signal value is smaller, a turn-off command is issued[2]. This switching action is very fast and introduces ripples in the output waveform of the inverter. Although a switched model of an inverter accurately describes these high frequency components present in the output waveform, the relationships between the modulating signal, which is the main control variable, and the corresponding current/voltage variables are not easily understood from the switched model. Also, for dynamic analysis purposes, knowledge of the high frequency components is often not necessary [2]. We are therefore concerned with the dynamics of the average values of the state variables, and as a result an average model for the three-phase inverter is developed (Figure 2.3) [2].

In Figure 2.3, $m_x$, $V_{tx}$, $i_x$ and $V_{sx}$ are the modulating signal, the inverter terminal voltage, the line current, and the grid voltage at phase $x$ respectively.

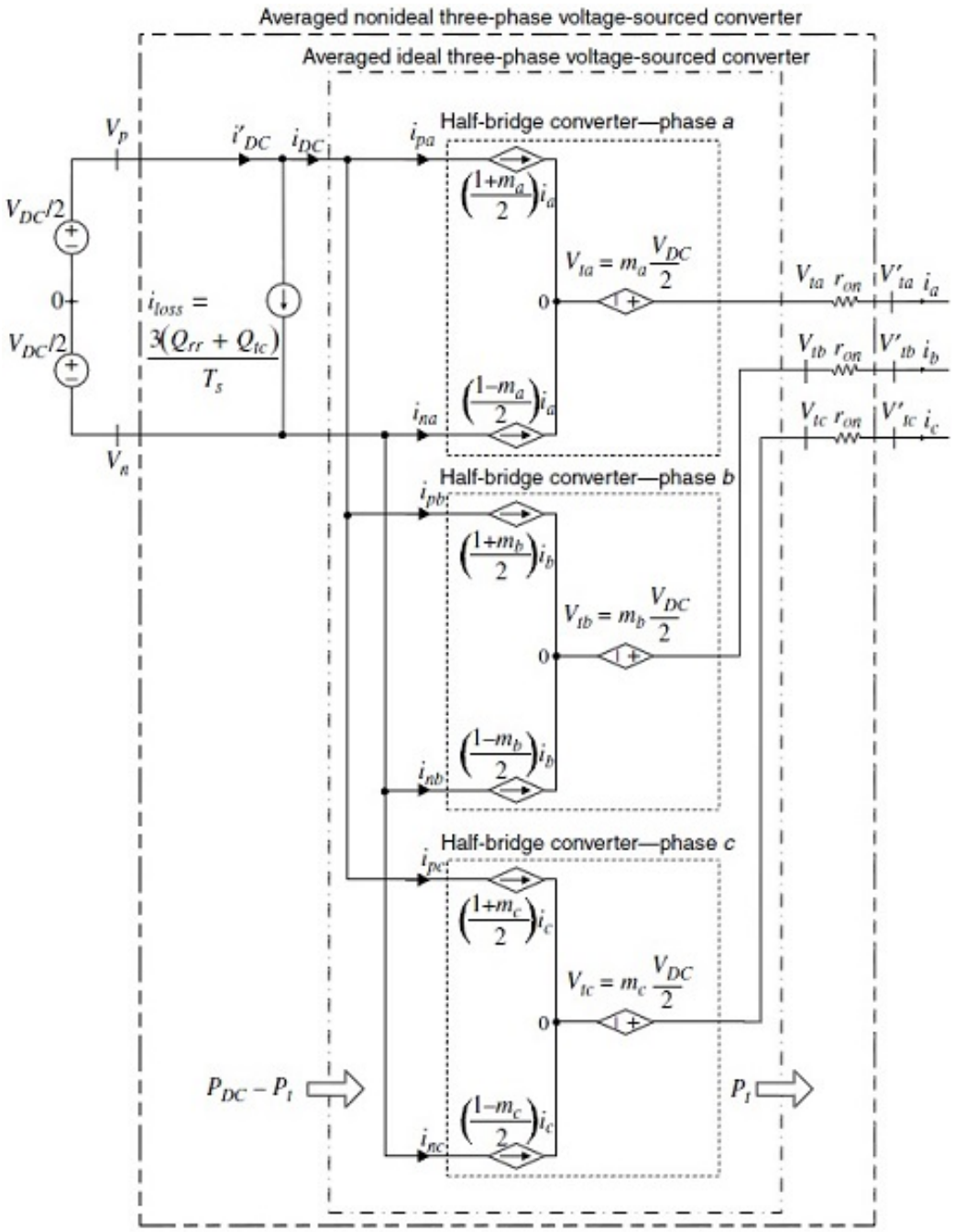


Figure 2.3: Averaged equivalent circuit of a three-phase voltage-sourced converter [2]

$\frac{3(Q_{rr}+Q_{tc})}{T_s}$ is the loss component of the input current, due to conduction and switching losses in the inverter [2].

### 2.2.1 Outer Power Control

The real power output of a PV system is dependent on the incident solar irradiance, the PV array temperature and the capacity of the PV arrays, and the reactive power output is determined by the inverter switching action [2]. Consequently, the real power output reference for inverter control is a function of the incident solar irradiance and the p-n junction temperature, and the reactive power output reference is determined by the system operator [18, 14]. Figure 2.4 shows a grid connected PV system.

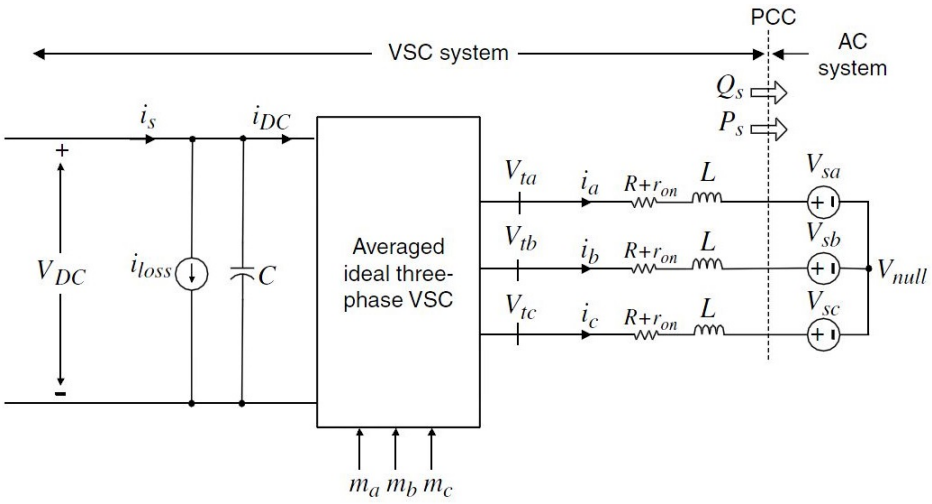


Figure 2.4: Grid-connected three-phase PV system model [2]

The point of common coupling (PCC) is where the inverter interfaces the grid. At the PCC, the following equations hold:

$$L\frac{di_a}{dt} = -(R + r_{on})i_a + V_{ta} - V_{sa} \tag{2.8}$$

$$L\frac{di_b}{dt} = -(R + r_{on})i_b + V_{tb} - V_{sb} \tag{2.9}$$

$$L\frac{di_c}{dt} = -(R + r_{on})i_c + V_{tc} - V_{sc} \tag{2.10}$$

The variables of the system described in equations (2.8)-(2.10) are sinusoidal quantities. Designing a control system for this system would involve a more complex compensator structure than if the variables were DC quantities [19]. Nonetheless, dq0 transformation of the system to a synchronously rotating reference frame [20] would transform these sinusoidal variables to DC quantities and reduce the number of subsystems from three to two, thus making analysis easier.

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos\theta & cos\left(\theta - \frac{2\pi}{3}\right) & cos\left(\theta + \frac{2\pi}{3}\right) \\ sin\theta & sin\left(\theta - \frac{2\pi}{3}\right) & sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \tag{2.11}$$

$$\theta = \int_0^t \omega(\epsilon)\, d\epsilon + \theta(0) \tag{2.12}$$

The dq0 transformation is expressed in equation (2.11). dq0 transformation of the PV system variables gives the following PCC equations:

$$L\frac{di_d(t)}{dt} = L\omega(t)i_q(t) - (R + r_{on})i_d(t) + V_{td}(t) - V_{sd}(t) \tag{2.13}$$

$$L\frac{di_q(t)}{dt} = -L\omega(t)i_d(t) - (R + r_{on})i_q(t) + V_{tq}(t) - V_{sq}(t) \tag{2.14}$$

$$\frac{d\theta(t)}{dt} = \omega(t) \tag{2.15}$$

The control variables $m_a$, $m_b$ and $m_c$ in Figure 2.3 may be expressed in the dq0 frame as:

$$V_{td}(t) = \frac{V_{DC}}{2} m_d(t) \tag{2.16}$$

$$V_{tq}(t) = \frac{V_{DC}}{2} m_q(t) \tag{2.17}$$

The instantaneous real and reactive power delivered to the AC system at the PCC are given as:

$$P(t) = \frac{3}{2}[V_{sd}(t)i_d(t) + V_{sq}(t)i_q(t)] \tag{2.18}$$

$$Q(t) = \frac{3}{2}[-V_{sd}(t)i_q(t) + V_{sq}(t)i_d(t)] \tag{2.19}$$

### 2.2.2 Phase Lock Loop

By synchronizing the rotating dq0 frame to the grid frequency ($\omega_o$) so that $V_{sq}(t) = 0$ and $\omega(t) = \omega_o$, we see that all the time varying quantities in equations (2.13)-(2.19) become constants, the power equations are reduced to equations (2.20) and (2.21) in steady state, and the real and reactive power controls are decoupled.

$$P(t) = \frac{3}{2}(V_{sd}i_d) \tag{2.20}$$

$$Q(t) = \frac{3}{2}(-V_{sd}i_q) \tag{2.21}$$

The control system for achieving this is called the phase lock loop (PLL), and a schematic is shown in Figure 2.5 [2]. Equation (2.23) expresses a mathematical model of the system in Figure 2.5. $K_P^{PLL}$ and $K_I^{PLL}$ are the proportional and integral gain constants of the compensator, and $\Phi_{LL} = \int_0^t V_{sq}dt$

$$\frac{d\theta(t)}{dt} = K_I^{PLL}\Phi_{LL} + K_P^{PLL}V_{sq} \tag{2.22}$$

$$\frac{d\Phi_{LL}}{dt} = V_{sq} \tag{2.23}$$

### 2.2.3 Inner Current Control

The system of equations (2.13)-(2.15) describe a non-linear system. To perform current control, feedback linearization is employed [19]. We define new control variables $U_d$ and $U_q$ as follows:

$$m_d = \frac{2}{V_{DC}}(U_d - L\omega_o i_q + V_{sd}) \tag{2.24}$$

$$m_q = \frac{2}{V_{DC}}(U_q + L\omega_o i_d + V_{sq}) \tag{2.25}$$

$$L\frac{di_d}{dt} = -(R + r_{on})i_d + U_d \tag{2.26}$$

$$L\frac{di_q}{dt} = -(R + r_{on})i_q + U_q \tag{2.27}$$

The resulting current control scheme is shown in Figure 2.6. The d-axis

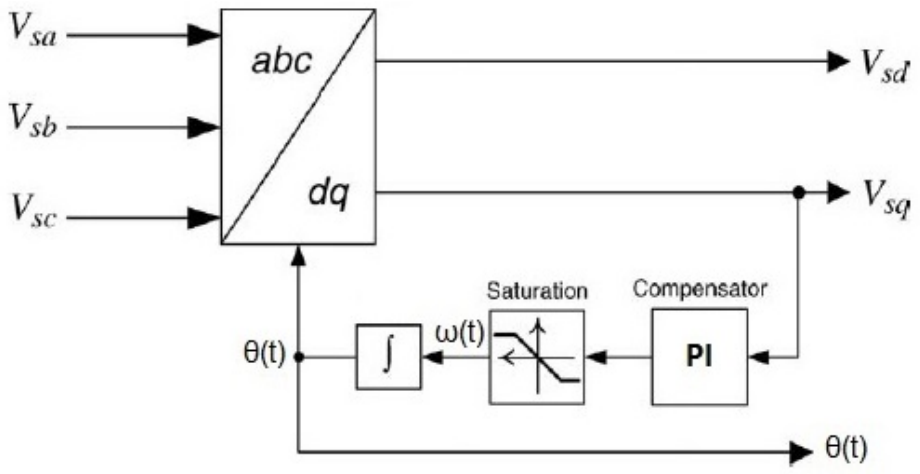


Figure 2.5: A phase lock loop control system [2]

and q-axis control plants are identical, so identical compensators may be used for the $d-frame$ control and the $q-frame$ control [2]. $K_P$ and $K_I$ are the proportional and integral gain constants of the compensators.

Equation (2.31) is a mathematical model of the system in Figure 2.6. The terms $i_{dref}$ and $i_{qref}$ are the reference values for d-axis and q-axis line currents respectively.

$$\frac{dq_d}{dt} = i_{dref} - i_d \tag{2.28}$$

$$\frac{dq_q}{dt} = i_{qref} - i_q \tag{2.29}$$

$$L\frac{di_d}{dt} = L\omega_o i_q - (R + r_{on})i_d + \frac{V_{DC}}{2}m_d(t) - V_{sd} \tag{2.30}$$

$$L\frac{di_q}{dt} = -L\omega_o i_d - (R + r_{on})i_q + \frac{V_{DC}}{2}m_q(t) - V_{sq} \tag{2.31}$$

The variables $m_d$ and $m_q$ are obtained from equations (2.24) and (2.25) re-

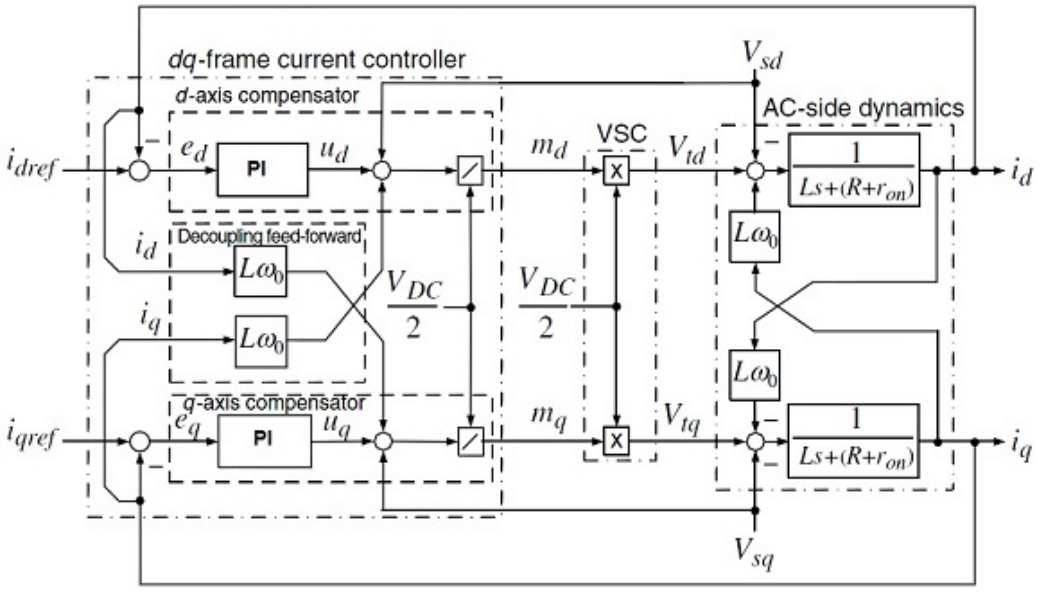


Figure 2.6: Current control block of a voltage sourced inverter [2]

spectively where:

$$U_d = K_P(i_{dref} - i_d) + K_I q_d \tag{2.32}$$

$$U_q = K_P(i_{qref} - i_q) + K_I q_q \tag{2.33}$$

Here, $i_{dref}$ and $i_{qref}$ are obtained from equations (2.18) and (2.19) respectively to be:

$$i_{dref} = \frac{2}{3}\left[\frac{V_{sd}P^{ref} + V_{sq}Q^{ref}}{V_{sd}^2 + V_{sq}^2}\right] \tag{2.34}$$

$$i_{qref} = \frac{2}{3}\left[\frac{-V_{sd}Q^{ref} + V_{sq}P^{ref}}{V_{sd}^2 + V_{sq}^2}\right] \tag{2.35}$$

### 2.2.4 Outer Voltage Control

By simply replacing the outer power control loop with an outer voltage control loop, the PV system can be made to control the PCC voltage and frequency [21]. In reference [22], a droop control strategy is used to determine

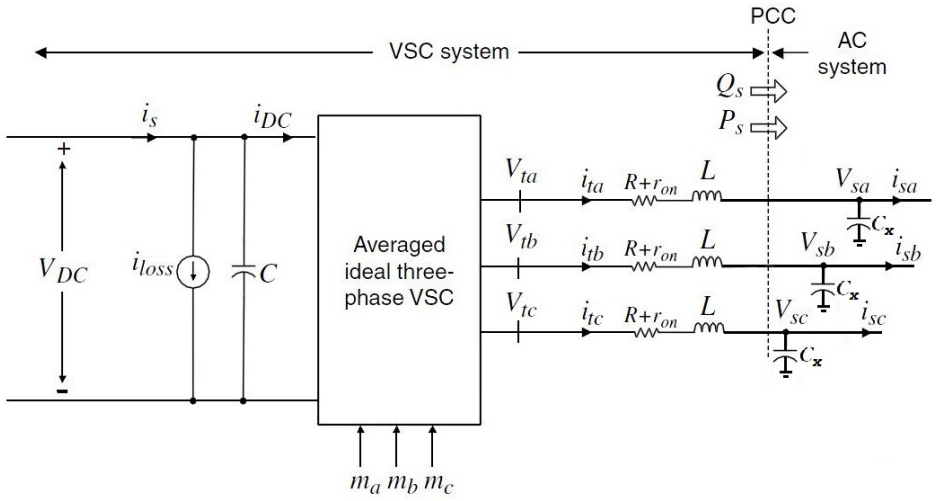


Figure 2.7: Autonomous three-phase PV system model

the voltage and frequency references for the outer voltage control loop. Based on the schematic in Figure 2.7, the following equations describe the dynamics at the PCC:

$$C_x \frac{dV_{sa}}{dt} = i_{ta} - i_{sa} \tag{2.36}$$

$$C_x \frac{dV_{sb}}{dt} = i_{tb} - i_{sb} \tag{2.37}$$

$$C_x \frac{dV_{sc}}{dt} = i_{tc} - i_{sc} \tag{2.38}$$

Transforming the variables to a dq0 reference frame rotating at reference frequency $\omega_{ref}$:

$$C_x \frac{dV_d(t)}{dt} = C_x \omega_{ref} V_{sq} + i_{td}(t) - i_{sd}(t) \tag{2.39}$$

$$C_x \frac{dV_q(t)}{dt} = -C_x \omega_{ref} V_{sd}(t) + i_{tq}(t) - i_{sq}(t) \tag{2.40}$$

$$\frac{d\delta_{pv}}{dt} = \omega_{ref} - \omega_s \tag{2.41}$$

and the following algebraic equations are satisfied at the PCC:

$$V_{pv}\cos\theta_{pv} = V_{sd}\cos\delta_{pv} - V_{sq}\sin\delta_{pv} \tag{2.42}$$

$$V_{pv}\sin\theta_{pv} = V_{sd}\sin\delta_{pv} + V_{sq}\cos\delta_{pv} \tag{2.43}$$

where $\delta_{pv}$ is the angular deviation of the PV system reference frame from the network reference frame, and $V_{pv}$ and $\theta_{pv}$ are the voltage magnitude and phase angle at the PV system bus respectively.

We introduce a proportional-integral (PI) controller for outer voltage loop control and employ feedback linearization. The dynamics of the outer voltage control are represented as follows:

$$\frac{dF_d}{dt} = V_{sdref} - V_{sd} \tag{2.44}$$

$$\frac{dF_q}{dt} = V_{sqref} - V_{sq} \tag{2.45}$$

$$U_{cd} = {K_P}^C(V_{sdref} - V_{sd}) + {K_I}^C F_d \tag{2.46}$$

$$U_{cq} = {K_P}^C(V_{sqref} - V_{sq}) + {K_I}^C F_q \tag{2.47}$$

The d-axis and q-axis control plants are identical, hence identical compensators may be used for the $d - frame$ control and the $q - frame$ control. ${K_P}^C$ and ${K_I}^C$ are the proportional and integral gain constants of the compensators respectively. The inner current loop reference points are obtained from the outer voltage loop through equation (2.49)

$$i_{tdref} = U_{cd} + i_{sd} - C_x\omega_{ref}V_{sq} \tag{2.48}$$

$$i_{tqref} = U_{cq} + i_{sq} + C_x\omega_{ref}V_{sd} \tag{2.49}$$

## 2.3 Synchronous Generator Model

A synchronous machine consists of a rotational member called the rotor or field and a stationary member referred to as the stator or armature. It is an alternating current (ac) machine whose rotational speed under steady-state conditions is proportional to the frequency in its armature. The magnetic field created by the armature currents rotates at the same speed as that created by field current on the rotor (which is rotating at synchronous speed),

and a steady torque results [23]. Synchronous generators or alternators are synchronous machines used to convert mechanical power to ac electric power [24]. References [20, 25] describe how the full model of synchronous generators is derived. This model consists of nine differential equations and three algebraic equations. To reduce model complexity, reduced order models are derived by approximating the behavior of fast dynamics without explicitly solving the associated differential equations. These are typically used in transient stability studies. Reduced order models for synchronous generators and their justifications are described in [26].

### 2.3.1 Two-Axis Model

In this work, the two-axis model of a synchronous generator is used. The two-axis model neglects the stator transients dynamics and the faster subtransient damper dynamics, and reduces the full model to four differential equations and four algebraic equations. When the dynamics of the exciter, the governor, the voltage regulator and the turbine are included, the two-axis model consists of nine differential equations and four algebraic equations [26]. The two-axis model for a synchronous generator used is shown in equations (2.50)-(2.62). The differential equations are:

$$T'_{do}\frac{dE'_q}{dt} = -E'_q - (X_d - X'_d)I_d + E_{fd} \tag{2.50}$$

$$T'_{qo}\frac{dE'_d}{dt} = -E'_d + (X_q - X'_q)I_q \tag{2.51}$$

$$\frac{d\delta}{dt} = \omega - \omega_s \tag{2.52}$$

$$\frac{2H}{\omega_s}\frac{d\omega}{dt} = T_M - E'_d I_d - E'_q I_q - (X'_q - X'_d)I_d I_q - D(\omega - \omega_s) \tag{2.53}$$

$$T_E\frac{dE_{fd}}{dt} = -(K_E + S_E(E_{fd}))E_{fd} + V_R \tag{2.54}$$

$$T_F\frac{dR_f}{dt} = -R_f + \frac{K_F}{T_F}E_{fd} \tag{2.55}$$

$$T_A\frac{dV_R}{dt} = -V_R + K_A R_f - \frac{K_A K_F}{T_F}E_{fd} + K_A(V_{ref} - V) \tag{2.56}$$

$$T_{CH}\frac{dT_M}{dt} = -T_M + P_{SV} \tag{2.57}$$

$$T_{SV}\frac{dP_{SV}}{dt} = -P_{SV} + P_C - \frac{1}{R_D}\left(\frac{\omega}{\omega_s} - 1\right) \tag{2.58}$$

The algebraic equations are:

$$E'_d - V\sin(\delta-\theta) - R_s I_d + X'_q I_q = 0 \tag{2.59}$$

$$E'_d - V\cos(\delta-\theta) - R_s I_q - X'_d I_d = 0 \tag{2.60}$$

$$I_d V\sin(\delta-\theta) + I_q V\cos(\delta-\theta) - P_L - P''_G = 0 \tag{2.61}$$

$$I_d V\cos(\delta-\theta) - I_q V\sin(\delta-\theta) - Q_L - Q''_G = 0 \tag{2.62}$$

The variables $E'_d$, $\omega$, $E_{fd}$, $T_M$ and $P_{SV}$ represent scaled flux linkage, generator speed, scaled field voltage, per unit electrical torque and per unit power output respectively. The variable $\delta$ represents the angular deviation of the synchronous generator reference frame from the network reference frame. The terms $X'$, $X$ and $S_E(E_{fd})$ are the scaled transient reactance, scaled reactance and saturation function respectively. The terms $R_D$, $H$ and $P_C$ are the droop coefficient, inertia constant and generator setpoint respectively. The term $P_L + jQ_L$ represents the complex power demand of a load at the generator bus, whereas the term $P''_G + jQ''_G$ represents the sum of the complex power supplied from the generator bus to other network buses and the line losses. Other variables and constants are defined in [26]

## 2.4 Network Model

The network model typically used for transient stability studies is employed. In this model, the network dynamics are neglected. This simplification stems from the fact that the time constants of rotating machines and their controls are much larger than that of the network [21].

Power flow algebraic equations are used to mathematically model the network. The power flow equations are non-linear and may be solved with iterative solution methods such as the Gauss-Siedel or Newton-Raphson methods [27]. The power flow equations for a $N$ bus power system are [27]:

$$P_{Gk} - P_{Lk} = V_k \sum_{n=1}^{N} V_n(G_{kn}\cos(\theta_k-\theta_n) + B_{kn}\sin(\theta_k-\theta_n) \tag{2.63}$$

$$Q_{Gk} - Q_{Lk} = V_k \sum_{n=1}^{N} V_n(G_{kn}\sin(\theta_k-\theta_n) - B_{kn}\cos(\theta_k-\theta_n) \qquad k = 1,2,\ldots,N \tag{2.64}$$

The terms $V_k$ and $\theta_k$ are the voltage magnitude and phase angle of bus $k$ respectively. The term $P_{Gk} + jQ_{Gk}$ is the total complex power supplied to bus $k$ by generators at bus $k$, and $P_{Lk} + jQ_{Lk}$ is the total complex power demand at bus $k$ from loads at bus $k$ The term $G_{kn} + jB_{kn}$ represents the row $k$ and column $n$ element of the bus admittance matrix ($\mathbf{Y_{bus}}$) described in [27].

## 2.5 Microgrid Reduced Order Model

Based on the models described in sections 2.1-2.4, a mathematical model for a microgrid integrated with PV generation and microturbines may be developed. The microturbine dynamics are modeled using equations (2.50)-(2.62), and the network is modeled using equations (2.63) and (2.64). When an outer voltage control is used, the PV system is modeled using equations (2.13)-(2.19), (2.24)-(2.33) and (2.39)-(2.49) while in contrast, the PV system is modeled using equations (2.13)-(2.35) when an outer power control is used. Equations (2.5) and (2.7) are used to compute maximum power point values for $V_{DC}$ and $I_{DC}$. The dynamics of a MPPT controller were not included in this analysis.

Upon careful inspection of these models, it is observed that the dynamics of the outer power control, the outer voltage control and the inner current control are on the order of the neglected network dynamics. Consequently, an integral manifold may be developed for the voltage and the currents by setting $\frac{dV_d}{dt} = \frac{dV_q}{dt} = \frac{dF_d}{dt} = \frac{dF_q}{dt} = 0$ and $\frac{di_d}{dt} = \frac{di_q}{dt} = \frac{dq_d}{dt} = \frac{dq_q}{dt} = 0$. This is equivalent to assuming that the output voltage, current and power are equal to the corresponding controller references by the next time step. This is a popular and valid assumption for microgrid analysis as cited in [28, 29, 30]

# CHAPTER 3

# POWER SHARING STRATEGIES FOR AUTONOMOUS MICROGRIDS

In autonomous microgrids, the units are responsible for the voltage control as well as the power sharing and balancing. The role of the power sharing feature is to ensure that all microsources share the load according to their ratings and availability of power from their energy source [3].

Existing power sharing strategies can be classified into communication based strategies and non-communication based strategies. The former requires a communication link between units for primary control and the latter does not.

## 3.1 Communication Based Strategies

### 3.1.1 Central Control/Concentrated Control

This strategy requires a communication link between a central controller and each unit in the system. The central controller maintains the balance in active power $P$ and reactive power $Q$ in steady-state conditions by coordinating the output of each unit in the microgrid. A major drawback of this approach is the cost of communication links and a supervisory control center in highly distributed and large systems [3].

### 3.1.2 Instantaneous(-Average) Current Sharing

This strategy requires a communication link between all the units. A current sharing bus and reference synchronization for the voltage is also needed. The voltage and current references of each unit are the shared information, with the objective of determining deviations of individual output current from desired values [31]. Since the output currents of the inverters are regu-

lated at every switching cycle, this strategy has a good performance both on current sharing and voltage regulation. Even if the output currents contain harmonics, power sharing can still be achieved. Nonetheless, the required interconnections between inverters limits the flexibility and redundancy of the system [3].

### 3.1.3 Master/Slave Control

A master is a unit that is responsible for system voltage control whereas a slave synchronizes to the system voltage and controls its current output. The master is also responsible for delivering the transient current and compensating for any mismatch between generation and load. In the master/slave control strategy, the master, or a central controller, specifies the reference currents for the slaves. Consequently, communication links are required between the master, or central controller, and the slaves. The drawbacks of this approach are the high communication bandwidth required as instantaneous values are communicated and the limits on system redundancy [3].

### 3.1.4 Distributed Control

This strategy requires a low bandwidth communication link between a central controller and the units. The voltage reference, the current reference and the averaged feedback voltage of all units are the shared signals. In the distributed control strategy, only the fundamental frequency components of the signals are communicated to the units from a central controller, whereas the higher frequency components are regulated to zero by a local controller [3]. The advantages of this approach include the low communication bandwidth required and the control quality it achieves. Nonetheless, its major drawback is the limit on system flexibility and expansion [3].

### 3.1.5 Angle Droop Control

This is similar to the method described in section 3.2.1. However, the angle droop control strategy requires a communication link or a GPS link for phase angle referencing. The phase angle, relative to the system-wide common

reference is used for the control [3]:

$$\delta = \delta_o - m(P - P_o) \tag{3.1}$$

$$V = V_o - n(Q - Q_o) \tag{3.2}$$

Although communication based strategies typically achieve good voltage regulation and power sharing while requiring simple inverter control algorithms, their major drawbacks are the high cost and vulnerability of communication lines, and the associated limits on system reliability, expandability and flexibility [3].

## 3.2 Non-Communication Based Strategies

Power sharing strategies that operate without inter-unit communication for the primary control are based on droop control [3]. These strategies have numerous advantages over the communication based strategies. For example, system expansion is easier, redundancy can be achieved easily, and the complexity, high cost and required reliability of a supervisory system are avoided [3]. Despite these advantages, some inherent drawbacks of droop control include the trade-off between power sharing accuracy and voltage deviations, imbalance in harmonic current sharing and dependency on the inverter output impedance [3]. Nonetheless some variations on the conventional droop controller are presented in reference [3] to overcome these disadvantages.

### 3.2.1 Droop Control

To achieve power sharing in the centralized grid, synchronous generators are equipped with a governor that ensures a mechanical-power/output-frequency droop. If the electrical power output of the generator is greater than the mechanical power output, the energy stored in the rotor makes up the difference, and the generator slows down. Since the rotational speed is proportional to the output frequency, the frequency of the terminal voltage also decreases (this corresponds to dynamically decreasing the terminal voltage phase angle) and because of transmission line characteristics, the electrical power output decreases. Each generator then measures its speed (which is directly

linked to the system frequency) and uses the power/frequency droop to adjust its mechanical power output. In this way, accurate power sharing between different generators is obtained [3]. Based on the network model presented

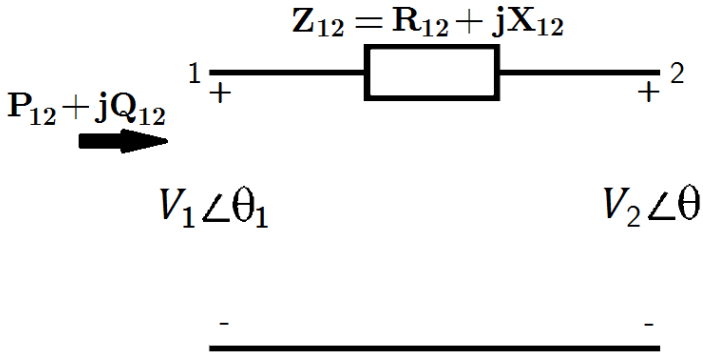


Figure 3.1: Power flow from bus 1 to bus 2

in section 2.4, the power flow equations for the system in Figure 3.1 are:

$$P_{12} = \frac{R_{12}V_1^2 - R_{12}V_1V_2\cos(\theta_1 - \theta_2) + X_{12}V_1V_2\sin(\theta_1 - \theta_2)}{Z_{12}^2} \tag{3.3}$$

$$Q_{12} = \frac{X_{12}V_1^2 - X_{12}V_1V_2\cos(\theta_1 - \theta_2) - R_{12}V_1V_2\sin(\theta_1 - \theta_2)}{Z_{12}^2} \tag{3.4}$$

Through a change of variables, the bus voltage magnitudes and angles can be expressed as:

$$V_2\sin(\theta_1 - \theta_2) = \frac{X_{12}P_{12} - R_{12}Q_{12}}{V_1} \tag{3.5}$$

$$V_1 - V_2\cos(\theta_1 - \theta_2) = \frac{R_{12}P_{12} + X_{12}Q_{12}}{V_1} \tag{3.6}$$

The following approximations are physically justifiable in a power system: $\cos(\theta_1 - \theta_2) \approxeq 1$ and $\sin(\theta_1 - \theta_2) \approxeq (\theta_1 - \theta_2)$ [22]. Therefore,

$$(\theta_1 - \theta_2) \approxeq \frac{X_{12}P_{12} - R_{12}Q_{12}}{V_1V_2} = Z_{12}P'_{12} \tag{3.7}$$

$$V_1 - V_2 \approxeq \frac{R_{12}P_{12} + X_{12}Q_{12}}{V_1} = Z_{12}Q'_{12} \tag{3.8}$$

where

$$\begin{bmatrix} P'_{12} \\ Q'_{12} \end{bmatrix} = \begin{bmatrix} \frac{X_{12}}{Z_{12}} & \frac{-R_{12}}{Z_{12}} \\ \frac{R_{12}}{Z_{12}} & \frac{X_{12}}{Z_{12}} \end{bmatrix} \begin{bmatrix} P_{12} \\ Q_{12} \end{bmatrix} \tag{3.9}$$

In transmission networks, $X_{ij} >> R_{ij}$, and this results in a loose coupling between the $P_{ij} - \Delta V$ and $Q_{ij} - \Delta\theta$ variables. Consequently, $P_{12}$ and $Q_{12}$ can be controlled by varying $\Delta\theta$ and $V$ respectively, or vice versa, and the conventional droop control strategy is based on this [32].

The conventional droop control strategy can be approached in two ways. In the first approach, each generator measures the system frequency and bus voltage, and adjusts its power output based on droop curves (Figures 3.2 and 3.3). During steady state, the frequencies of all generators match the system frequency, the generator bus voltages are equal, and power sharing is achieved; equations (3.10) and (3.11) show the droop control equations. In the second approach, each generator measures its instantaneous power output and adjusts its output voltage frequency and magnitude according to droop curves (Figures 3.2 and 3.3). When steady state is reached, the frequencies and voltage magnitudes of each generator match, and power sharing is achieved. This method cannot be employed on synchronous generators because the output frequency is linked to the rotational speed and cannot be controlled independently; even so, it is common in converter-based microgrids [3]. The droop control equations are expressed in (3.12) and (3.13):

$$P = PC_P - \frac{1}{R_D}(\frac{\omega}{\omega_s} - 1) \tag{3.10}$$

$$Q = PC_Q - \frac{1}{R_Q}(V - 1) \tag{3.11}$$

or

$$\omega = \omega_s + R_D\omega_s(PC_P - P) \tag{3.12}$$

$$V = 1 + R_Q(PC_Q - Q) \tag{3.13}$$

The terms $PC_P$ and $PC_Q$ are control inputs that can either be constant, or obtained from a central controller for adjusting real and reactive power setpoints respectively. The constants $R_D$ and $R_Q$ are the droop coefficients, and the variable $V$ is the per unit generator bus voltage.

In distribution networks, the line resistances are significant relative to line reactances, and the same approximation cannot be made. Nonetheless, by performing a linear transformation on $P_{ij}$ and $Q_{ij}$, new variables can be defined (3.9) and a modified droop control strategy formulated [22].

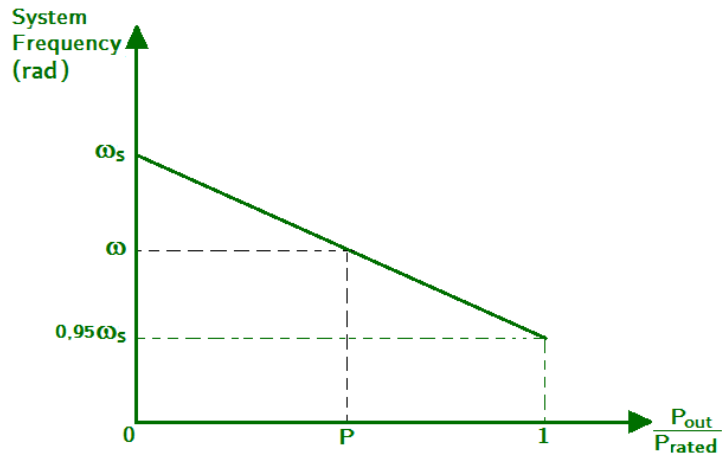


Figure 3.2: Frequency droop characteristic (5% droop, $PC_P = 0$)

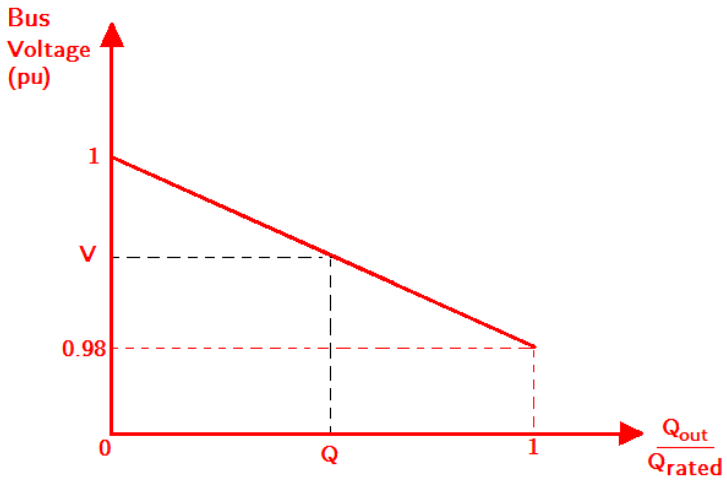


Figure 3.3: Voltage droop characteristic (2% droop, $PC_Q = 0$)

Droop Control + Central Controller

A disadvantage of the droop control strategy is that although the steady state frequencies and generator bus voltages match, they typically deviate from the nominal values. Nonetheless, a central controller can be used to restore the system frequency and generator bus voltages to their nominal values [32]. The central controller measures the steady state system frequency, and communicates new setpoints to each unit. This requires that a low bandwidth communication link exist between the central controller and each unit.

## 3.3 Area Control Error

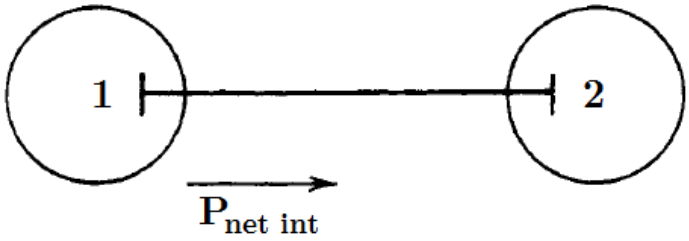


Figure 3.4: A two-area system

A control area is a part of an interconnected system within which the load and generation will be controlled according to the rules in Figure 3.4. The control area's boundary is simply the tie-line points where power flow is metered [33]. The total control area net interchange $P_{net\ int}$ must be monitored.

The area control error (ACE) is a centralized generation concept. It represents the shift in an area's generation required to restore system frequency and net interchange to their desired values [33]. Based on the droop control equations in (3.10)-(3.13) and the method described in reference [33], the following equations can be used to compute the area control error $ACE_P$, and an equivalent term we define as $ACE_Q$ for the reactive power.

$$ACE_P = -\Delta P_{net\ int} - \frac{(\omega - \omega_s)}{R_D \omega_s} \tag{3.14}$$

$$ACE_Q = -\frac{V - 1}{R_Q} \tag{3.15}$$

A central controller computes the $ACE_P$ and $ACE_Q$ for each area and communicates new generation setpoints so each area adjusts its generation accordingly.

# CHAPTER 4

# MODEL DEVELOPMENT

This chapter introduces a power sharing based algorithm for optimizing PV generation. The strategy, founded on concepts introduced in earlier chapters, allows PV systems participate in frequency and voltage regulation within their capability. The algorithm is designed for autonomous microgrids with an integration of PV generators and microturbines, and a 5-bus test system is studied.

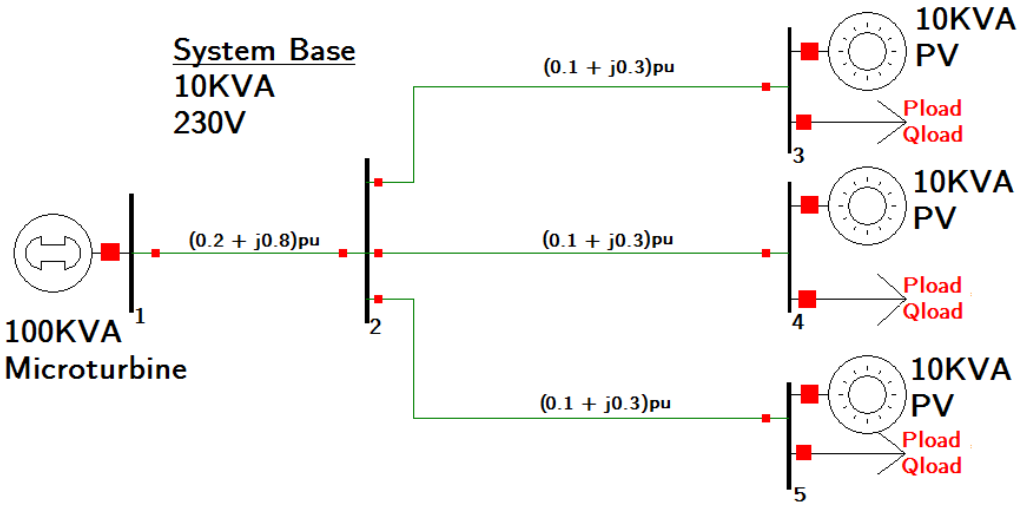


Figure 4.1: Autonomous microgrid: A 5-bus test system

## 4.1 Five-Bus Test System

The system in Figure 4.1 represents an autonomous microgrid integrated with three PV generators, a microturbine, and three constant power loads, with the significant line resistance present in distribution networks accounted for. To test the proposed strategy analytically, a mathematical model first has to be developed. We therefore formulate a mathematical model for the

5-bus test system that is based on the models discussed in Chapter 2.

The microturbine is modeled using the two-axis model described in section 2.3. The dynamics of the PV generators are neglected (section 2.5), and the network is modeled using equations (2.63) and (2.64). The microturbine dynamics are modeled as follows:

$$T'_{do}\frac{dE'_q}{dt} = -E'_q - (X_d - X'_d)I_d + E_{fd} \tag{4.1}$$

$$T'_{qo}\frac{dE'_d}{dt} = -E'_d + (X_q - X'_q)I_q \tag{4.2}$$

$$\frac{d\delta}{dt} = \omega - \omega_s \tag{4.3}$$

$$\frac{2H}{\omega_s}\frac{d\omega}{dt} = T_M - E'_d I_d - E'_q I_q - (X'_q - X'_d)I_d I_q - D(\omega - \omega_s) \tag{4.4}$$

$$T_E\frac{dE_{fd}}{dt} = -(K_E + S_E(E_{fd}))E_{fd} + V_R \tag{4.5}$$

$$T_F\frac{dR_f}{dt} = -R_f + \frac{K_F}{T_F}E_{fd} \tag{4.6}$$

$$T_A\frac{dV_R}{dt} = -V_R + K_A R_f - \frac{K_A K_F}{T_F}E_{fd} + K_A(V_{ref} - V_1) \tag{4.7}$$

$$T_{CH}\frac{dT_M}{dt} = -T_M + P_{SV} \tag{4.8}$$

$$T_{SV}\frac{dP_{SV}}{dt} = -P_{SV} + P_C - \frac{1}{R_D}\left(\frac{\omega}{\omega_s} - 1\right) \tag{4.9}$$

The algebraic equations are:

$$E'_d - V_1\sin(\delta - \theta_1) - R_s I_d + X'_q I_q = 0 \tag{4.10}$$

$$E'_d - V_1\cos(\delta - \theta_1) - R_s I_q - X'_d I_d = 0 \tag{4.11}$$

$$\begin{aligned} I_d V_1\sin(\delta - \theta_1) + I_q V_1\cos(\delta - \theta_1) - \\ (G_{11}V_1^2 + V_1V_2(G_{12}\cos(\theta_1 - \theta_2) + B_{12}\sin(\theta_1 - \theta_2))) = 0 \end{aligned} \tag{4.12}$$

$$\begin{aligned} I_d V_1\cos(\delta - \theta_1) - I_q V_1\sin(\delta - \theta_1) - \\ (-B_{11}V_1^2 + V_1V_2(G_{12}\sin(\theta_1 - \theta_2) - B_{12}\cos(\theta_1 - \theta_2))) = 0 \end{aligned} \tag{4.13}$$

The network model is expressed below. At bus 2:

$$V_2 \sum_{n=1}^{5} V_n(G_{2n}\cos(\theta_2 - \theta_n) + B_{2n}\sin(\theta_2 - \theta_n) = 0 \tag{4.14}$$

$$V_2 \sum_{n=1}^{5} V_n(G_{2n}\sin(\theta_2 - \theta_n) - B_{2n}\cos(\theta_2 - \theta_n) = 0 \tag{4.15}$$

and at buses 3, 4 and 5:

$$-P_{PVk} + P_{Lk} + G_{kk}V_k^2 + V_kV_2(G_{k2}\cos(\theta_k - \theta_2) + B_{k2}\sin(\theta_k - \theta_2)) = 0 \tag{4.16}$$

$$-Q_{PVk} + Q_{Lk} - B_{kk}V_k^2 - V_kV_2(G_{k2}\sin(\theta_k - \theta_2) - B_{k2}\cos(\theta_k - \theta_2)) = 0 \tag{4.17}$$

$$k = 3, 4, 5$$

$P_{PVk} + jQ_{PVk}$ represents the output complex power of the PV generator at bus k, and $P_{Lk} + jQ_{Lk}$ represents the load at bus k. The simultaneous implicit (SI) method introduced in [26] is employed for dynamic simulation.

## 4.2 Power Sharing Strategy

Of all the power sharing strategies presented in Chapter 3, the droop control + central controller strategy shows more advantages than the others. Although this makes it a seemingly best approach for the autonomous microgrid in Figure 4.1, the time scale disparity between PV system and microturbine dynamics makes the strategy difficult to implement on the test system (however, in reference [34], a method for adjusting inverter control dynamics to mimic that of synchronous generators is discussed, and these inverters are called synchronverters). We therefore develop a modified master slave strategy for power sharing.

### 4.2.1 Modified Master/Slave Strategy

The master/slave strategy presented in section 3.1.3 involves units dedicated for system voltage and frequency control, and slaves that synchronize to the

bus voltage and system frequency and act as scheduled generation units. In this strategy, the reference powers for the slaves are specified by a central controller or the master, and as a result a communication link is required between them.

In the modified master/slave strategy, the masters and slaves still play these roles, but the reference powers for the slaves are determined locally through a droop control strategy. Each slave measures the system frequency and bus voltages locally, and adjusts its output real and reactive power according to specified droop curves. In this way, the slaves contribute to frequency and bus voltage regulation as well as power sharing using locally measured information. By introducing a central controller, the system frequency and generator bus voltages can be restored to nominal values through low bandwidth communication links. An advantage of this approach over synchronverters is that it takes advantage of the fast response of PV generators. To implement the modified master/slave control strategy on the five-bus test bus, the microturbines are masters and the PV generators are slaves. The PV generators synchronize to the bus voltage and system frequency through a phase lock loop, and an outer power loop control is employed (section 2.2.1). The modified master/slave control strategy is depicted in Figure 4.2.

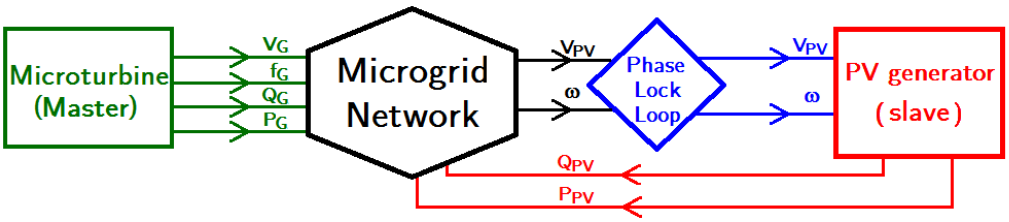


Figure 4.2: Modified master/slave control strategy

To obtain real power references for the outer power loops, each PV generator droops its real power output and computes an internal frequency ($\omega_{int}$) which is used to determine the real power reference according to equation (4.18). This process is repeated until the internal frequency matches the measured value, and at this stage real power sharing is achieved. The frequency droop is depicted in Figure 4.3.

$$P_{PVk}^{ref} = PC_{Pk} - \frac{1}{R_D}\left(\frac{\omega + \omega_{int\,k}}{2\omega_s} - 1\right) \tag{4.18}$$

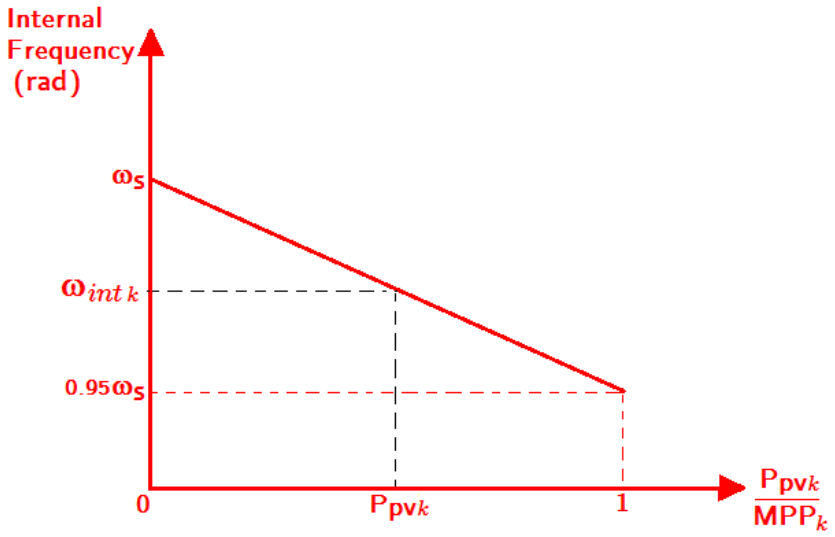


Figure 4.3: PV generator (slave) frequency droop (5% droop, $PC_{Pk} = 0$)

To obtain reactive power references for the outer power loops, each PV generator droops the linear transformed reactive power ($Q'_{PV}$) introduced in equation (3.9) and computes an internal voltage ($V_{int}$) which is used to determine the reactive power reference according to equations (4.19), (4.20) and (4.21). This process is repeated until the internal voltage matches the measured value, and at this stage reactive power sharing is achieved. The voltage droop is depicted in Figure 4.4

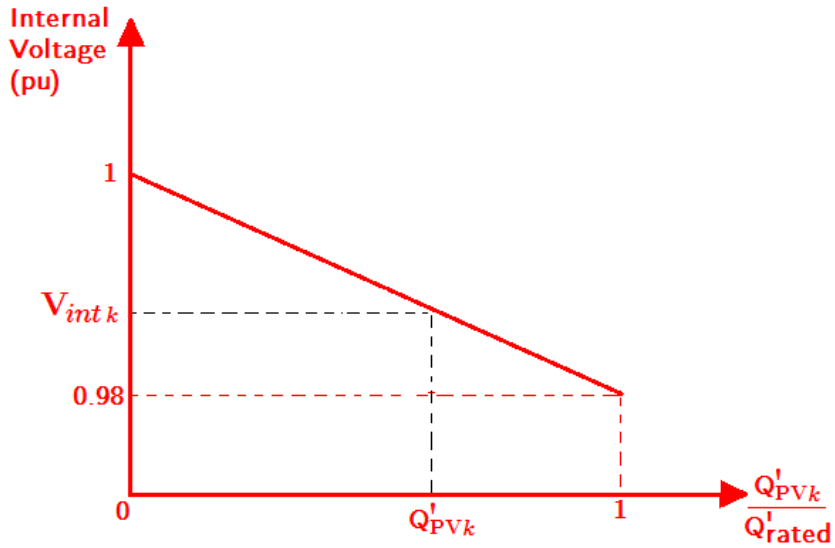


Figure 4.4: PV generator (slave) voltage droop (2% droop, $PC_{Qk} = 0$)

$$Q_{PVk}^{ref\,\prime} = PC_{Qk} - \frac{1}{R_Q}(\frac{V + V_{int\,k}}{2} - 1) \tag{4.19}$$

$$Q_{PVk}^{ref\,*} = \frac{Z_{k2}}{X_{k2}} Q_{PVk}^{ref\,\prime} - \frac{R_{k2}}{X_{k2}} P_{PVk} \tag{4.20}$$

Here, k2 is the line between bus $k$ and bus 2.

The fast response of the PV system caused instability in the voltage magnitude control, and to avoid this a delay was introduced in the voltage droop of the PV system as follows:

$$T_{Qk} \frac{dQ_{PVk}^{ref}}{dt} = -Q_{PVk}^{ref} + Q_{PVk}^{ref\,*} \tag{4.21}$$

Note that the PV generators droop $P_{PV}$ with frequency and $Q_{PV}^{'}$ with bus voltage. This is because the system frequency is controlled by a rotating machine whose speed can be regulated by adjusting the electrical power output, and the significant line resistances in the distribution system make bus voltages proportional to $Q_{PV}^{'}$, not $Q_{PV}$.

## 4.3 Area Control Strategy

With the modified master/control strategy presented in the previous section, power sharing can be achieved among units according to their capabilities. Nonetheless, the strategy does not solve the problem of optimizing PV generation. To achieve this, we propose an area control strategy (ACS). The ACS involves splitting the autonomous microgrid into two areas: one composed of microturbines and other non-renewable generation, and the other containing renewable generation and loads as depicted in Figure 4.5. The idea is that by minimizing the area interchange $P_{line}$ within generator capabilities, we optimize PV generation.

The goal is to optimize PV generation while maintaining power quality. To achieve this, we employ the concept introduced in section 3.3, and for this to work, each PV generator must communicate its MPP to the central controller as it changes. The central controller measures the system frequency and the bus voltage of a PV generator. Combining this with the MPP of each generator, the area control error which minimizes $P_{line}$ is computed and

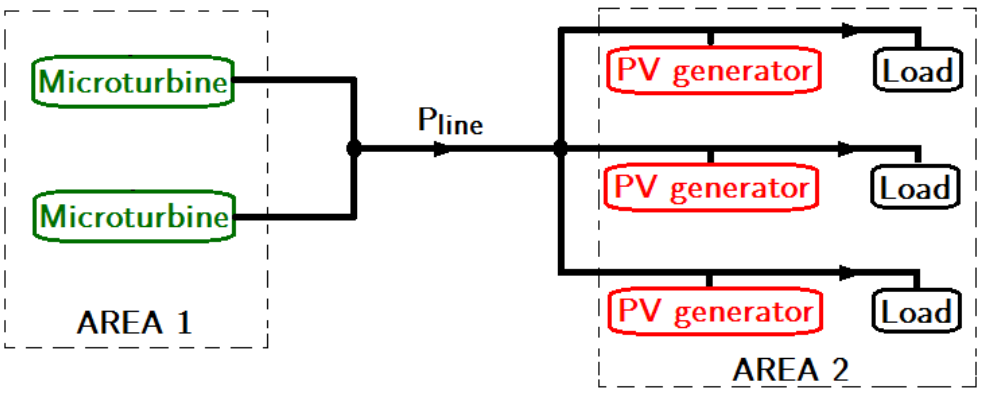


Figure 4.5: Area control strategy

used to determine new setpoints for each unit within their capabilities.

$$\Delta P_{PV}^{max} = \sum_k MPP_k - \sum_k P_{PVk} \tag{4.22}$$

$$ACE_{P1}^1 = -\min(\Delta P_{PV}^{max}, P_{line}) - \frac{(\omega - \omega_s)}{R_D \omega_s} \tag{4.23}$$

$$ACE_{Pk}^2 = \frac{MPP_k \min(\Delta P_{PV}^{max}, P_{line})}{\sum_k MPP_k} - \frac{(\omega - \omega_s)}{R_D \omega_s} \tag{4.24}$$

$$ACE_{Qk}^2 = -\frac{V_k - 1}{R_Q} \tag{4.25}$$

$MPP_k$ and $P_{PVk}^{ref}$ are the maximum power point and output power reference of the PV generator at bus k respectively. $ACE_{Pk}^m$ is the area control error for a generator at bus $k$ in area $m$. New setpoints for the generators in the system are determined as follows:

$$PC = PC + ACE_{P1}^1 \tag{4.26}$$

$$PC_{Pk} = PC_{Pk} + ACE_{Pk}^2 \tag{4.27}$$

$$PC_{Qk} = PC_{Qk} + ACE_{Qk}^2 \tag{4.28}$$

The new setpoints minimize the area interchange and restore the system frequency and bus voltages to nominal values. As a result, available PV generation can be optimized while maintaining power quality. Figure 4.6 shows how the ACS strategy works with the modified master/slave strategy to optimize PV generation.

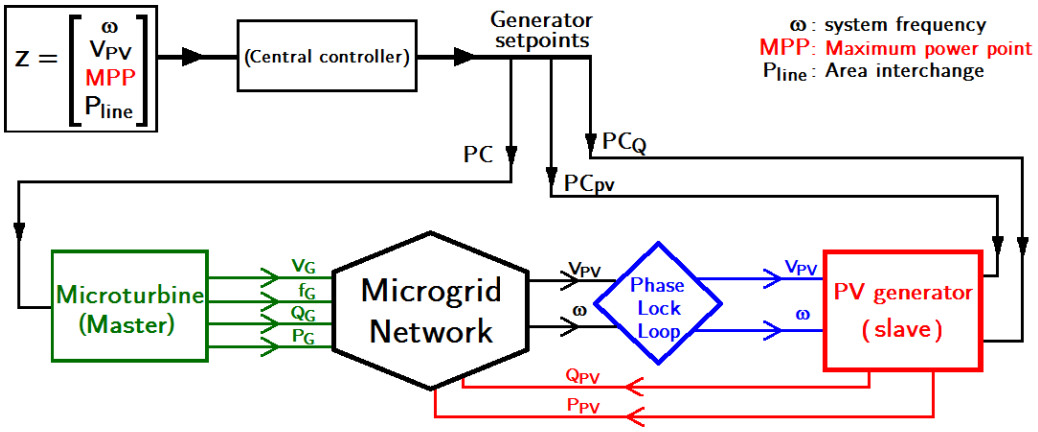


Figure 4.6: Area control strategy (ACS) + modified master/slave control

Microturbine turn-off

By combining information on $\Delta P_{PV}^{max}$ and $P_{line}$ with load forecasting, the central controller can predict when PV generation is sufficient to meet the system demand. Consequently, the microturbine can be turned off to better optimize PV generation and eliminate the cost of having the microturbines control the system frequency. To achieve this, the PV generator must be controlled to operate as masters by changing the outer power loop to an outer voltage loop as described in section 2.2.4. The droop control strategy is then combined with the ACS. Figure 4.7 shows how the ACS optimizes PV generation when the microturbine is turned off.

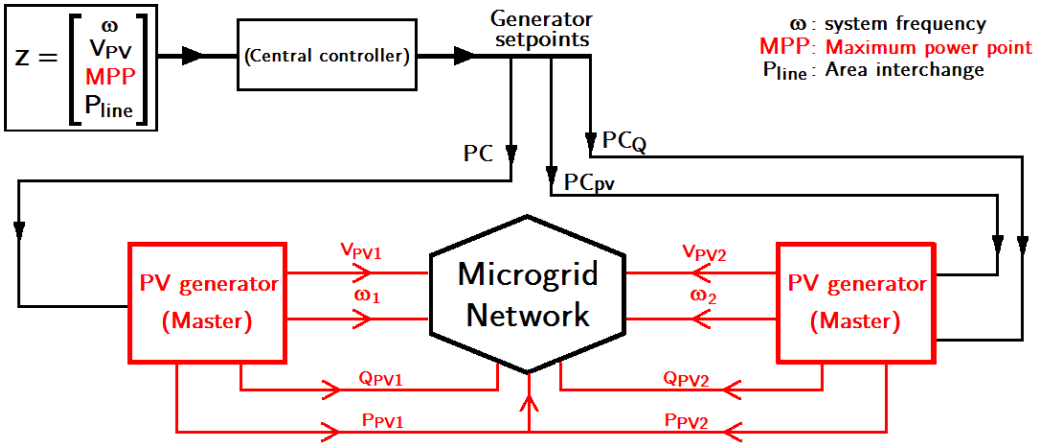


Figure 4.7: Area control strategy (ACS) + droop control

The second approach to droop control mentioned in section 3.2.1 is used, and since the line resistances are significantly large, a linear transformation of

the output real and reactive powers (equation (3.9)) is drooped to determine new output voltage magnitudes and frequencies for each unit. In this way, real and reactive power sharing is achieved [22].

In the simulation, the control system predicts the load demand variation and if the total MPP of all PV units at a time of day is three times greater than the forecasted demand, the central controller issues a turn-off command to the microturbine and an outer-loop switch command to the PV generators.

## 4.4 Energy Storage System Integration

The proposed power sharing strategy performs power curtailment by operating the PV generators below MPP when necessary. Although this feature reduces the effect of sharp increases in MPP on power quality, the effect of sharp decreases in MPP are not contained since the PV system cannot operate above MPP. Nonetheless, by integrating an energy storage system (ESS) into the PV system, it can operate above MPP and the effect of sharp MPP decrease abated. The ESS is integrated as shown in Figure 4.8.

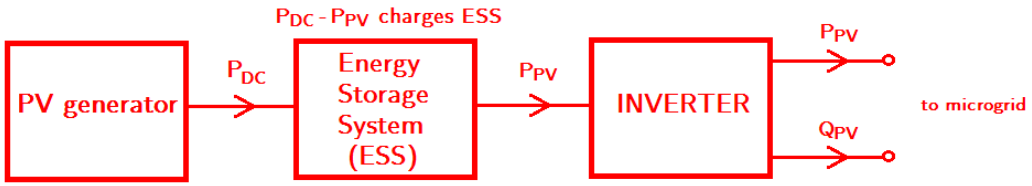


Figure 4.8: PV generator + ESS

To modify the strategy, the variable $MPP_k$ is replaced with $BattMPP_k$ in the modified master/slave control equations. This new variable is equal to the maximum power point, but a limit is placed on $\left|\frac{dBattMPP_k}{dt}\right|$. By placing this limit on the rate of increase or decrease of $BattMPP_k$ in the control equations, sharp changes in MPP for each PV generator are buffered and the energy storage systems make up the difference in PV generator output and power delivered to the microgrid. Also, since the new setpoints for each PV system are chosen based on the MPP, the central controller is operated to reset this variable so that $BattMPP_k = MPP_k$ at the start of every control cycle.

# CHAPTER 5

# RESULTS AND DISCUSSION

The autonomous microgrid in Figure 5.1, with parameters summarized in appendix A, is modeled in Matlab. The simulation is run over a 12 hour period, from sunrise to sunset, with a time step 100 milliseconds. The central controller cycle is 10 seconds. Two case studies are investigated.

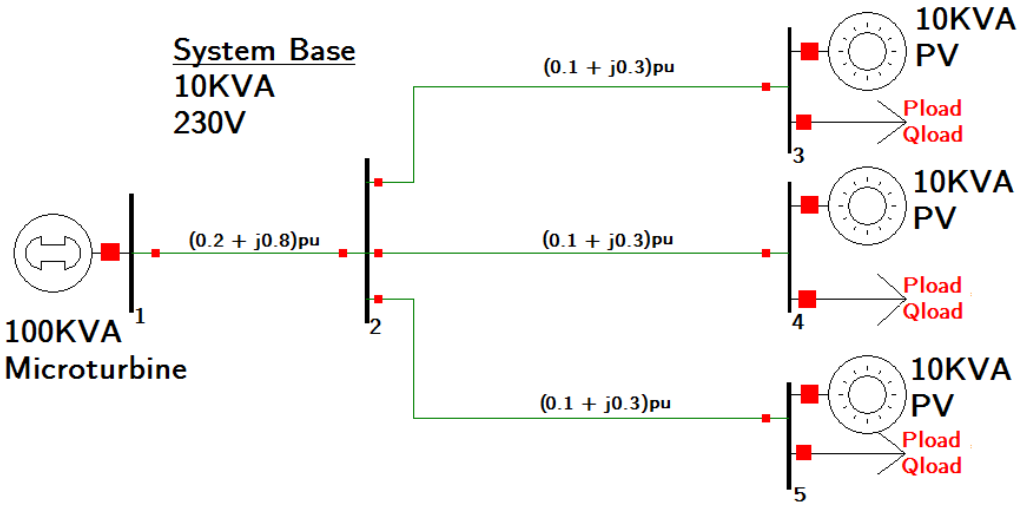


Figure 5.1: Autonomous microgrid: A 5-bus test system

## 5.1 Without Cloud Cover or Shading

In the first case study, the effects of cloud cover and shading are neglected, and motivated by reference [15], the MPP from sunrise to sunset is modeled as a bell-shaped variation. Perfect MPPT is assumed, and the MPP is tracked every 100 milliseconds. The same constant power load model is assumed for all the loads. The inputs to the system are the MPP and the loads, and the system response is studied. Figures 5.2 - 5.10 show the simulation results for this case study.

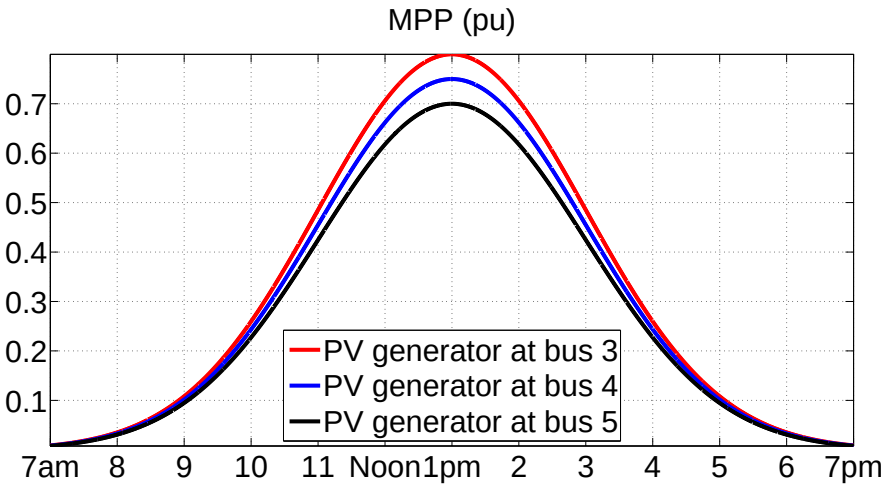


Figure 5.2: Maximum power point of PV generators (No cloud cover/shading)

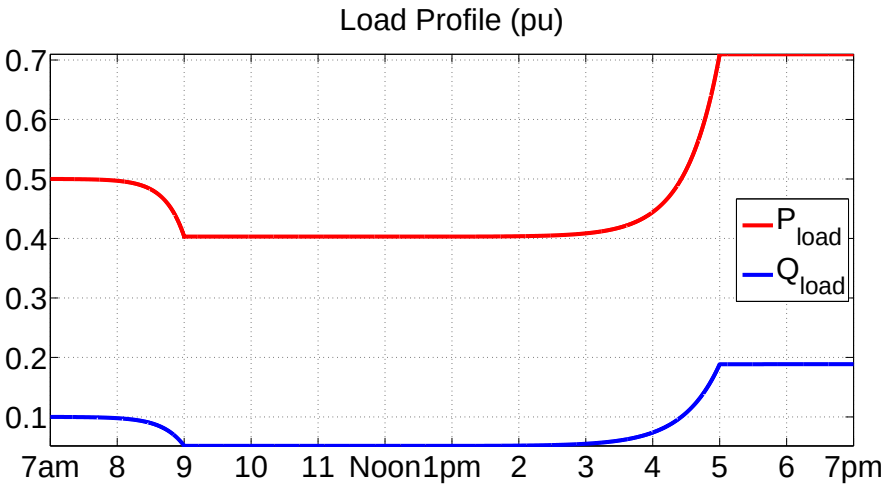


Figure 5.3: Load profile for $P_{load}$ and $Q_{load}$ (no cloud cover/shading)

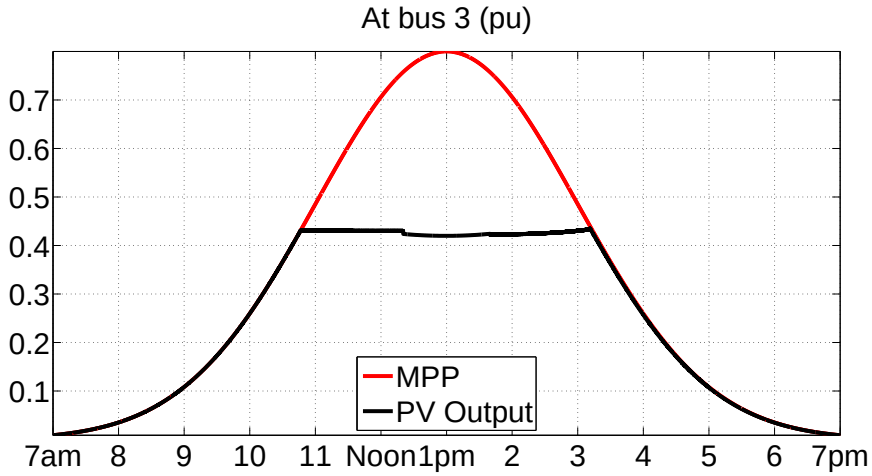


Figure 5.4: $P_{PV3}$ vs. $MPP_3$ (no cloud cover/shading)

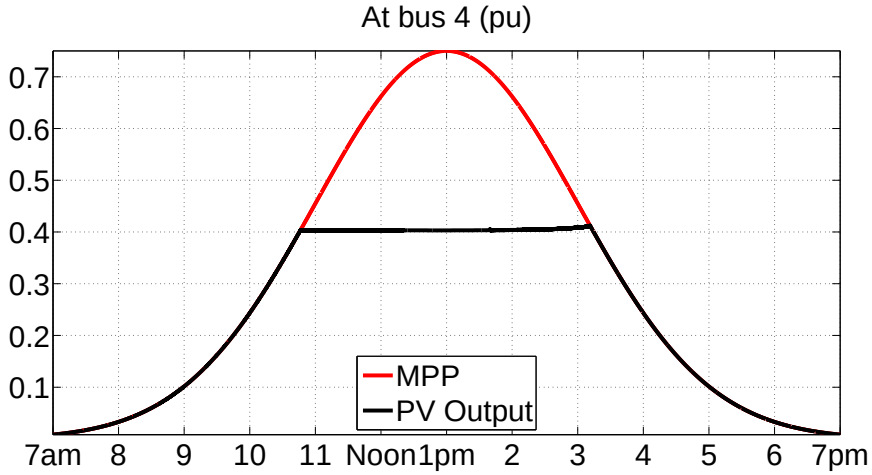


Figure 5.5: $P_{PV4}$ vs. $MPP_4$ (no cloud cover/shading)

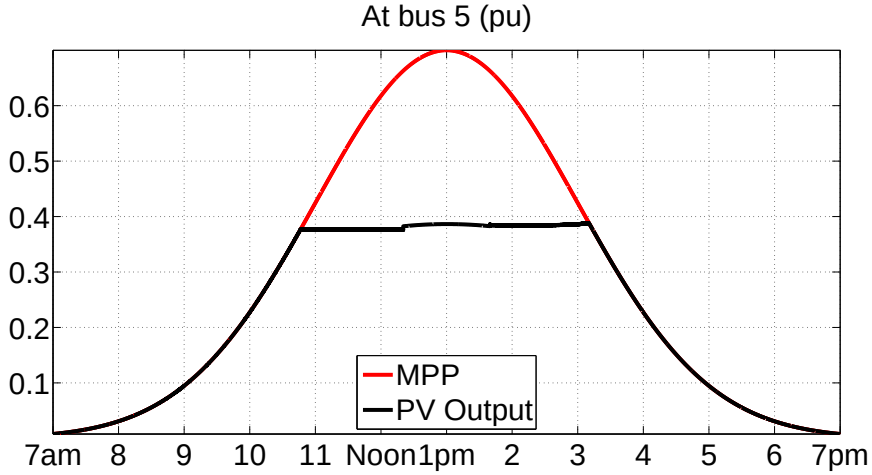


Figure 5.6: $P_{PV5}$ vs. $MPP_5$ (no cloud cover/shading)

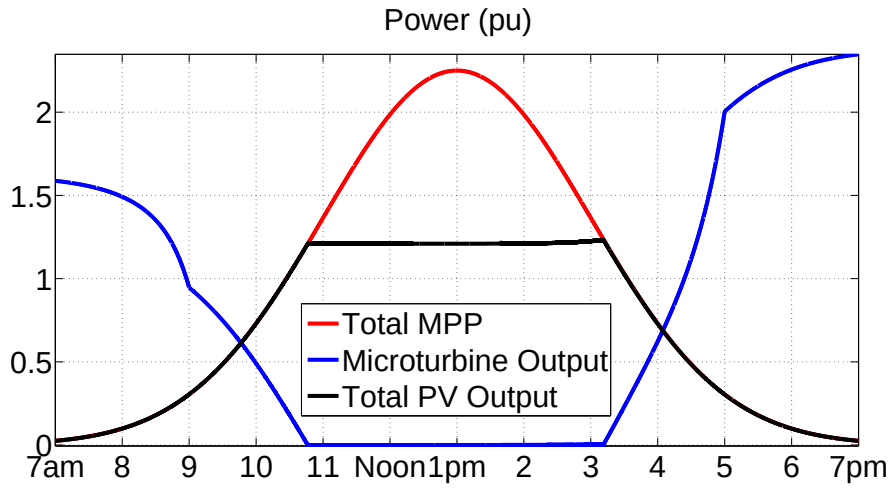


Figure 5.7: Total PV generation vs. microturbine output (no cloud cover/shading)

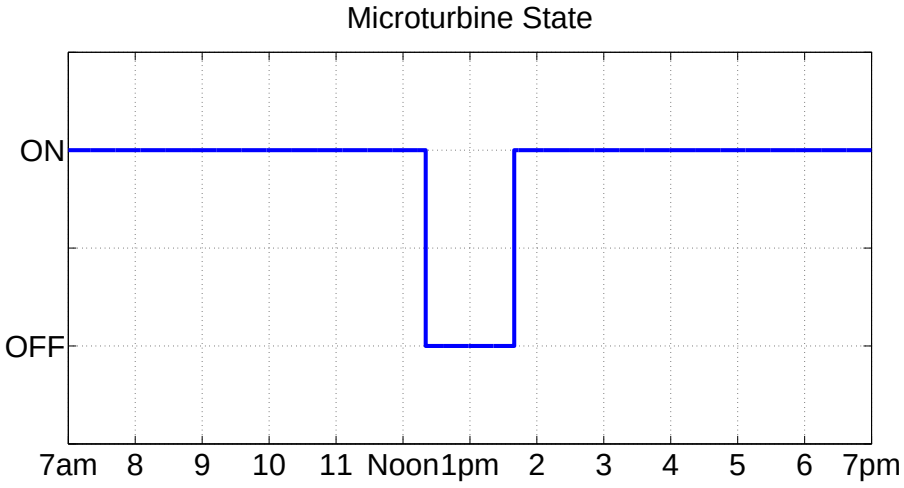


Figure 5.8: Microturbine state (ON or OFF) (no cloud cover/shading)

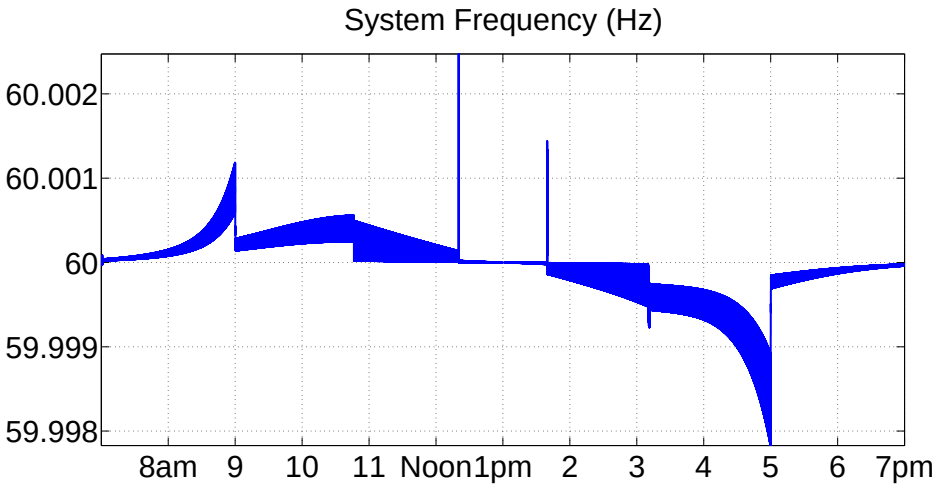


Figure 5.9: System frequency (Hz) (no cloud cover/shading)

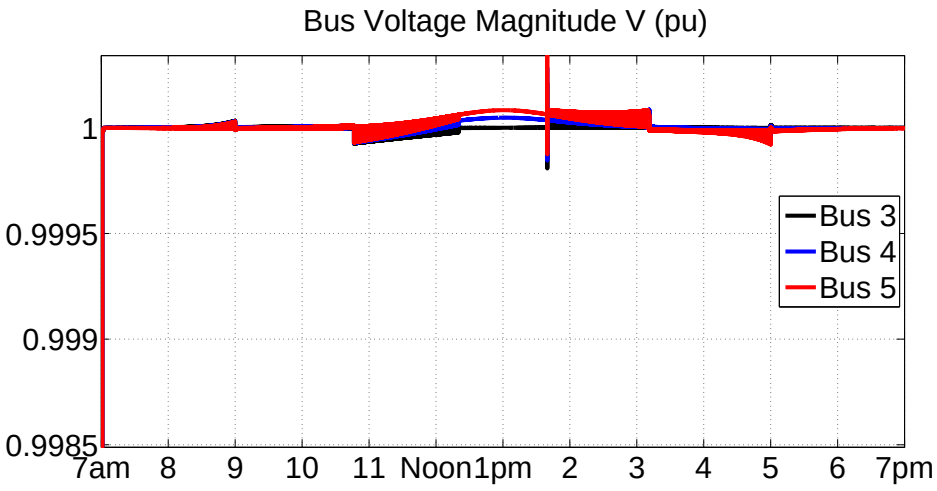


Figure 5.10: Bus voltages (pu) (no cloud cover/shading)

## 5.2 With Cloud Cover and Shading

In this case study, the effects of cloud cover and shading are included in the MPP models for each PV unit.

Cloud cover is modeled as a homogeneous Markov chain with four states from 0% cloud cover to 70% cloud cover, at increments of 17.5%. The transitions are modeled to occur every five minutes, and the state transition probabilities are inversely proportional to the difference between the states: for example, the probability of transitioning from 35% cloud cover to 17.5% cloud cover is higher than the probability of transitioning to 70% cloud cover. The effect of shading is modeled as a homogeneous Markov chain with sixteen states from 0% reduction to 15% reduction, at increments of 1%. The transitions are modeled to occur every 100 milliseconds, and the state transition probabilities are inversely proportional to the difference between the states.

As in the previous case study, perfect MPPT is assumed, and the MPP is tracked every 100 milliseconds. The same constant power load model is also assumed for all the loads. Figure 5.3 shows the load profile for this case study, and Figures 5.11 - 5.19 show the simulation results.

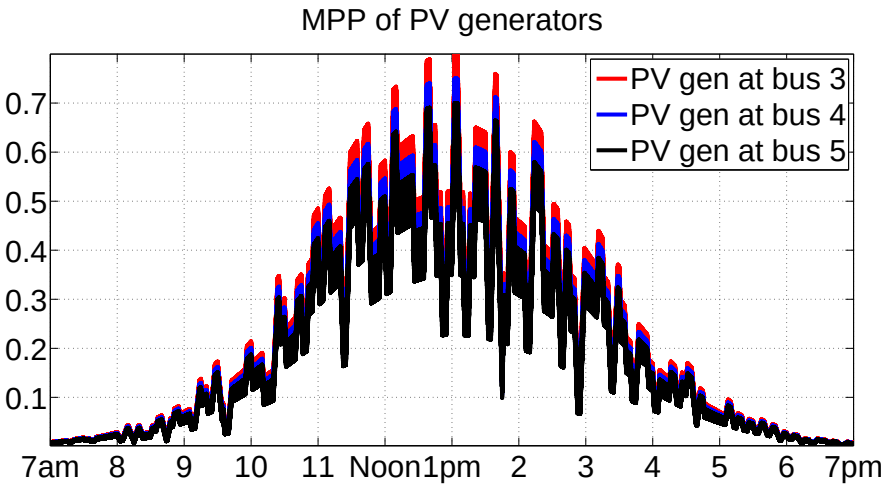


Figure 5.11: Maximum power point of PV generators (with cloud cover and shading)

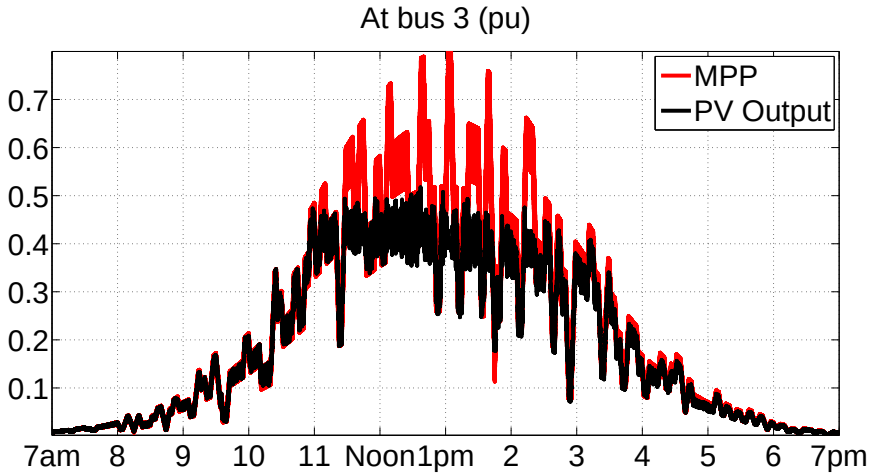


Figure 5.12: $P_{PV3}$ vs. $MPP_3$ (with cloud cover and shading)

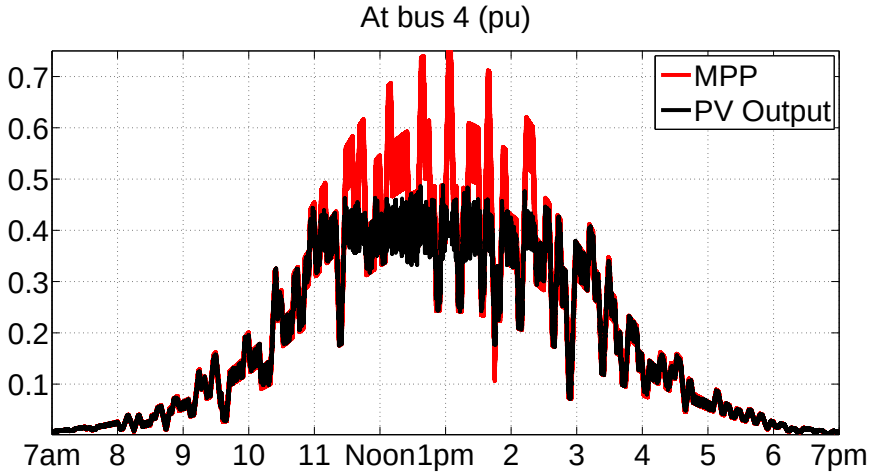


Figure 5.13: $P_{PV4}$ vs. $MPP_4$ (with cloud cover and shading)

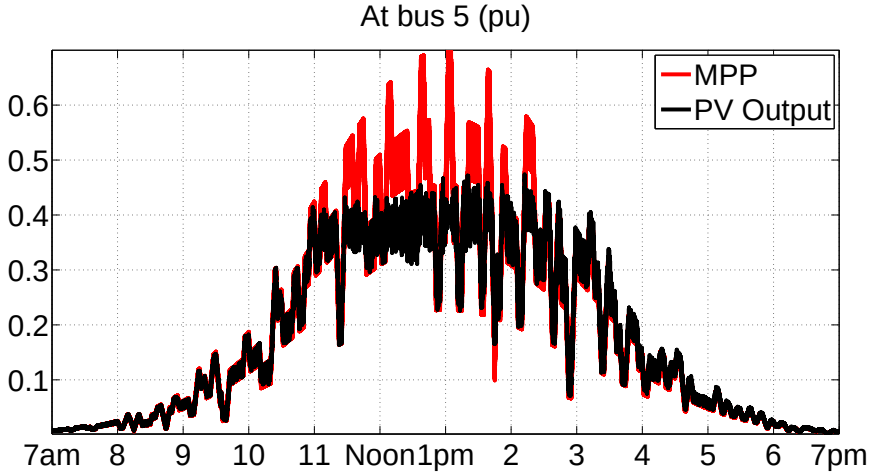


Figure 5.14: $P_{PV5}$ vs. $MPP_5$ (with cloud cover and shading)

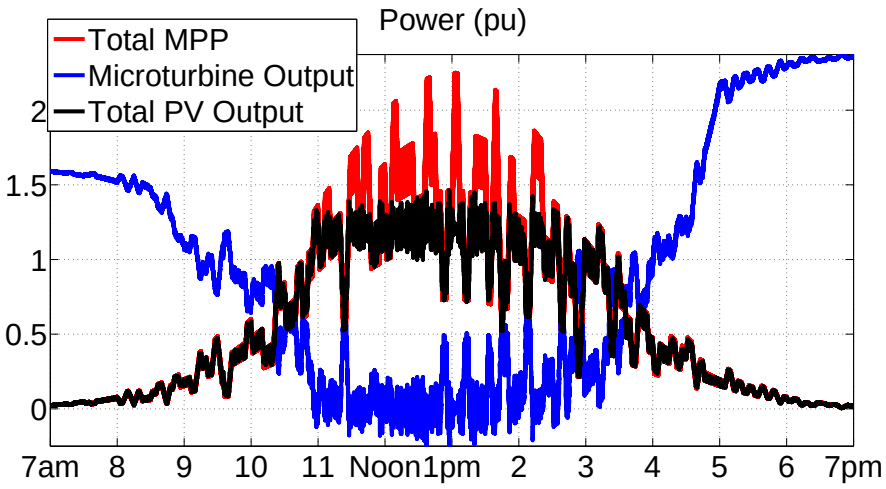


Figure 5.15: Total PV generation vs. microturbine output (with cloud cover and shading)

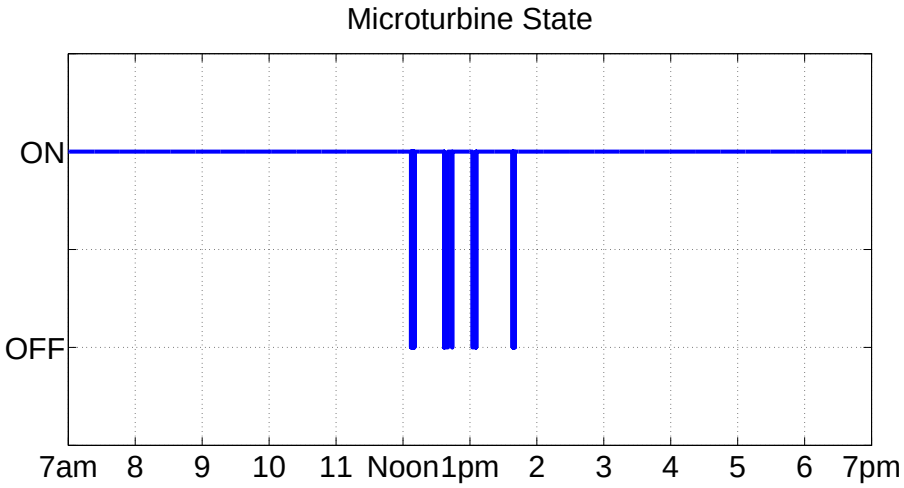


Figure 5.16: Microturbine state (ON or OFF) (with cloud cover and shading)

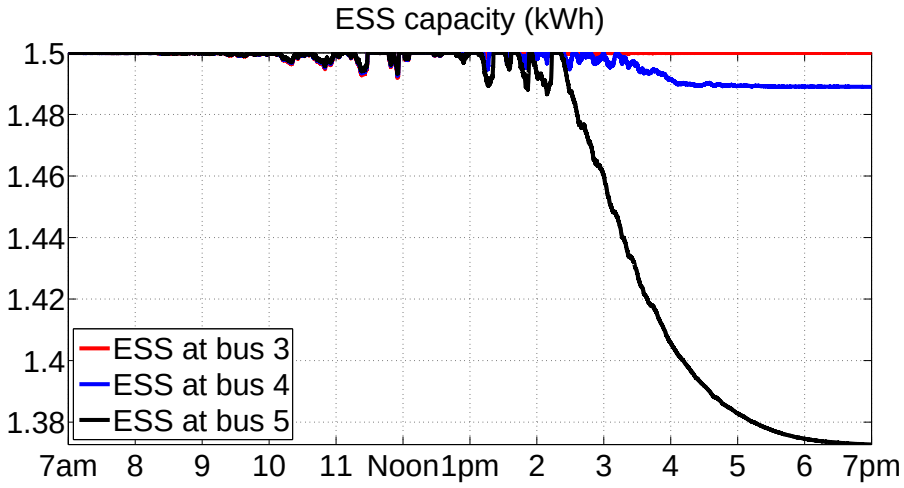


Figure 5.17: ESS capacity for PV systems

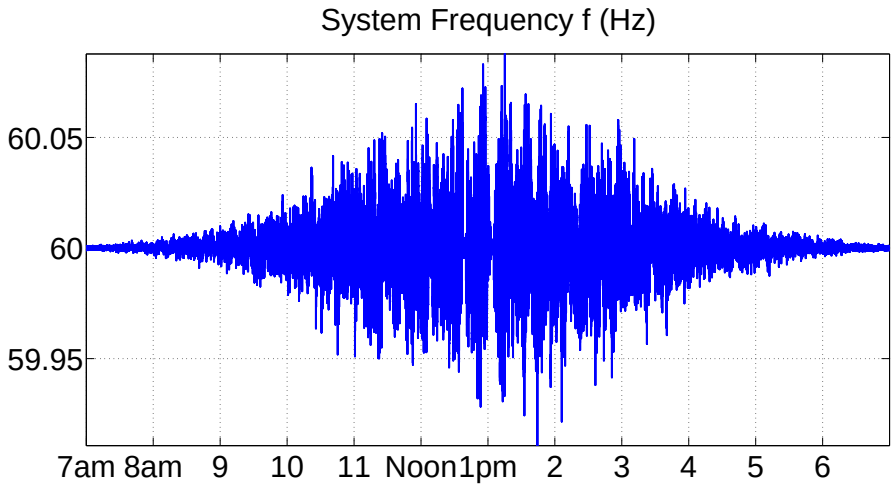


Figure 5.18: System frequency (Hz) (with cloud cover and shading)

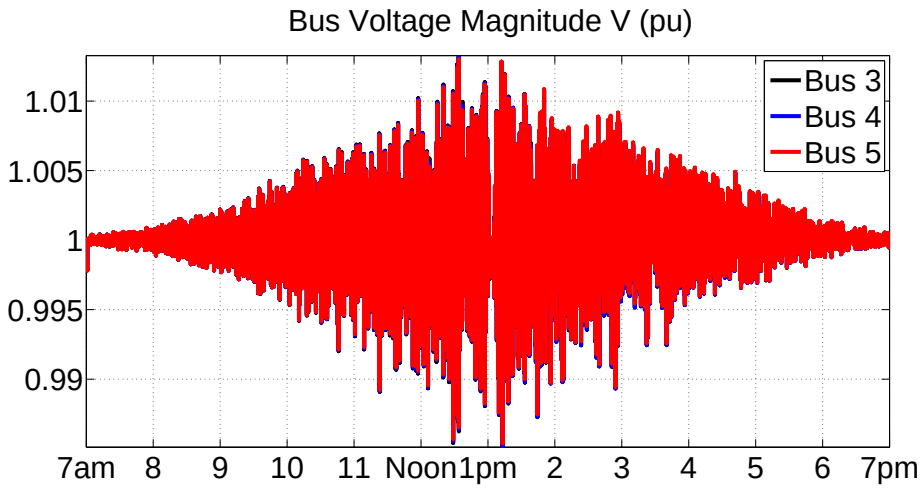


Figure 5.19: Bus voltages (pu) (with cloud cover and shading)

## 5.3 Discussion

The first case study involves an ideal case of no PV generation intermittency and the MPP of each PV generator is modeled to vary according to Figure 5.2. Figures 5.4 - 5.6, show that the MPP is tracked until around 10:45 a.m. when the MPP exceeds the system demand. At this stage, the microturbine electrical power output is zero (Figure 5.7), so the PV generators supply the system demand. Figure 5.8 shows that the microturbine is turned off at around 12:15 p.m. (one hour and 30 minutes after PV generation is sufficient). This is caused by inaccurate load forecasting in the control system. Figures 5.9-5.10, show that the system frequency and bus voltages are maintained close to the nominal values of 60 Hz and 1 pu respectively.

The second case study involves an extreme case of severe PV generation intermittency where the MPP drops or increases by 85% in five minutes. The MPP of each PV generator is modeled to vary according to Figure 5.11. Figures 5.12 - 5.14 show that despite the intermittency of MPP, the control system is able to track the MPP during hours of generation capacity, and Figure 5.16 shows that the microturbine is turned off during periods of high generation capacity relative to the forecasted demand. Figure 5.19 shows that the bus voltages are maintained close to the nominal value 1 pu, and is well within power quality specifications of $< 1.05$ pu and $> 0.95$ pu. On the other hand, Figure 5.18 shows that the frequency exceeds power quality specifications of $< 60.05$ Hz and $> 59.95$ Hz. This results from the extreme intermittency considered, and although the control system excursion from power quality specifications is observed to occur for less than 5 minutes, the results can be further improved by increasing the system inertia.

# CHAPTER 6

# CONCLUSION

A power sharing based strategy for optimizing available PV generation in autonomous microgrids was presented and time domain simulations used to verify its effectiveness. The analysis began with setting up models for the test system, and then developing a control strategy for sharing the system demand among generators according to their capabilities. Afterwards, an area control strategy was introduced for optimizing the available PV generation. The importance of integrating an ESS into the PV system was then highlighted, and the control equations were modified accordingly. The simulations were performed in Matlab, and the results showed that for extreme PV generation intermittency, the bus voltages were well within power quality specifications. However, system frequency excursions from power quality specifications were observed, and this was attributed to the low system inertia.

## 6.1 Future Work

The next phase of this work involves testing the proposed strategy on a standardized test system and considering other renewable energy resources. For a more practical analysis of the control strategy, testing with experimental MPP data and other load models will also be explored.

# APPENDIX A

# MICROGRID PARAMETERS

This appendix summarizes the microgrid system parameters. Table A.1 shows the parameters used in the simulation.

Table A.1: These parameters correspond to the test system in Figure 5.1 and the corresponding models developed in Chapter 4

| | |
|---|---|
| $H(seconds)$ | 23.64 |
| $X_d(pu)$ | 0.146 |
| $X_d^{'}(pu)$ | 0.0608 |
| $X_q(pu)$ | 0.0969 |
| $X_q^{'}(pu)$ | 0.0969 |
| $T_{do}^{'}(seconds)$ | 8.96 |
| $T_{qo}^{'}(seconds)$ | 0.31 |
| $K_A$ | 20 |
| $T_A(seconds)$ | 0.2 |
| $K_E$ | 1.0 |
| $T_E(seconds)$ | 0.314 |
| $K_F$ | 0.063 |
| $T_E(seconds)$ | 0.35 |
| $S_E(E_{fd})$ | $0.00394\exp^{1.555E_{fd}}$ |
| $R_{12}(pu)$ | 0.2 |
| $X_{12}(pu)$ | 0.8 |
| $R_{23}(pu)$ | 0.1 |
| $X_{23}(pu)$ | 0.3 |
| $R_{24}(pu)$ | 0.1 |
| $X_{24}(pu)$ | 0.3 |
| $R_{25}(pu)$ | 0.1 |
| $X_{25}(pu)$ | 0.3 |
| $T_{Qk}(seconds)$ | 8 |

# APPENDIX B

# AUTONOMOUS MICROGRID SIMULATION CODE

This appendix contains the Matlab simulation code used to model the five-bus test system in Figure 5.1.

## B.1 Microgrid System

```
clc
clear
%Zero PV power (early hours of the day): All load is supplied by the
...synchronous generator


tic

%System parameters


global  H D Rs Xd Xdp Xq Xqp Tdop Tqop...    %Machine data
        KA TA KE TE KF TF...                 %Exciter Data
        TSV TCH RD...                        %Speed governor data
        Vg angle_g wg ws Pload3 Qload3 Pload4 Qload4 Pload5 Qload5 Pg Qg Edp delta Eqp...
        G11 B11 G22 B22 G33 B33 G44 B44 G55 B55 G12 B12 G21 B21 G23 B23 G32 B32 G24 B24 G42 B42 G25
```

```
            B25 G52 B52...
17          MPP3 MPP4 MPP5 Qrated3 Qrated4 Qrated5 P_pv3
                Q_pv3 P_pv4 Q_pv4 P_pv5 Q_pv5 V_pv3
                angle_pv3...
18          w_pv3 V_pv4 angle_pv4 w_pv4 V_pv5 angle_pv5
                w_pv5 V2 angle2 i...
19          h...
20          C_pcc...
21          Ibase_bus1 Zbase_gen Zbase_bus1 Zbase_net
                Ibase_gen Sgen_RATED Spv_RATED3 Spv_RATED4
                Spv_RATED5 n_pv3 n_pv4 n_pv5
22
23  %Step size
24  hrs=12;
25  secs=3600*hrs;
26  MPPtime=0.1;
27
28  h = 0.1;                                        %step size
29
30  datapoints=secs/h;
31  MPPdatapoints=(secs/MPPtime);
32  MPPintervals=datapoints/MPPdatapoints;
33
34  MPPxxx=zeros(1,MPPdatapoints+1);
35
36  m = datapoints+1;  %time from 7am to 7pm is 43200
        seconds (step size of h)
37
38  loadfunction
39
40                              Vg = zeros(1,m);
41                          angle_g = zeros(1,m);
42                              Pg = zeros(1,m);
43                              Qg = zeros(1,m);
44                              I_g=zeros(1,m);
45                              V_DC =1000*ones(1,m);
```

```
46          V2 = zeros(1,m);
47          V_pv3 = zeros(1,m);
48          V_pv4 = zeros(1,m);
49          V_pv5 = zeros(1,m);
50          angle2 = zeros(1,m);
51          angle_pv3 = zeros(1,m);
52          angle_pv4 = zeros(1,m);
53          angle_pv5 = zeros(1,m);
54          w_pv3 = zeros(1,m);
55          w_pv4 = zeros(1,m);
56          w_pv5 = zeros(1,m);
57          SlowCloudState = zeros(1,m);
58          MPP3 = zeros(1,m);
59          BattMPP3 = zeros(1,m);
60          Qrated3 = zeros(1,m);
61          MPP4 = zeros(1,m);
62          BattMPP4 = zeros(1,m);
63          Qrated4 = zeros(1,m);
64          MPP5 = zeros(1,m);
65          BattMPP5 = zeros(1,m);
66          Qrated5 = zeros(1,m);
67          P_PVDC3 = zeros(1,m);
68          B3_CAPACITY = zeros(1,m);
69          P_pv3 = zeros(1,m);
70          Q_pv3 = zeros(1,m);
71          P_PVDC4 = zeros(1,m);
72          B4_CAPACITY = zeros(1,m);
73          P_pv4 = zeros(1,m);
74          Q_pv4 = zeros(1,m);
75          P_PVDC5 = zeros(1,m);
76          B5_CAPACITY = zeros(1,m);
77          P_pv5 = zeros(1,m);
78          Q_pv5 = zeros(1,m);
79          Edp = zeros(1,m);
80          Eqp = zeros(1,m);
81          delta = zeros(1,m);
```

```
82                                  wg = zeros(1,m);
83                                  w_sys = zeros(1,m);
84                                  Efd = zeros(1,m);
85                                  Rf = zeros(1,m);
86                                  VR = zeros(1,m);
87                                  TM = zeros(1,m);
88                                  PSV = zeros(1,m);
89                                  delta_pv3=zeros(1,m);
90                                  Vsd=zeros(1,m);
91                                  Vsq=zeros(1,m);
92                                  m_pv3 = zeros(1,m);
93                                  m_pv23 = zeros(1,m);
94                                  n_pv3 = zeros(1,m);
95                                  m_pv4 = zeros(1,m);
96                                  m_pv24 = zeros(1,m);
97                                  n_pv4 = zeros(1,m);
98                                  m_pv5 = zeros(1,m);
99                                  m_pv25 = zeros(1,m);
100                                 n_pv5 = zeros(1,m);
101                                 V_pv_ref3 = zeros(1,m);
102                                 V_pv_ref4 = zeros(1,m);
103                                 V_pv_ref5 = zeros(1,m);
104                                 Turn_off_SG = zeros(1,m);
105                                 DELTA_Ppv_max = zeros(1,m);
106
107 %Network
108
109 Sgen_RATED=100e3;
110 NET_BASE=10e3;
111 %VA rating of synchronous generator
112 Vgen_nominal=400; %nominal generator bus voltage
113 Ibase_gen=Sgen_RATED/Vgen_nominal;     %base generator
        current
114 Zbase_gen=Vgen_nominal/Ibase_gen; %base generator
        impedance
115
```

```
116 %Load at bus 3
117 Pload_bus3=7e3; %7kW
118 Qload_bus3=2e3; %6kVAr
119
120 %Load at bus 4
121 Pload_bus4=7e3; %7kW
122 Qload_bus4=2e3; %6kVAr
123
124 %Load at bus 5
125 Pload_bus5=7e3; %7kW
126 Qload_bus5=2e3; %6kVAr
127
128 %400V:230V Transformer between buses 1 and 2
129 Pbase=10e3; %network base power
130
131 %base parameters at bus 1
132 Vbase_bus1=400;
133 Ibase_bus1=Pbase/Vbase_bus1;
134 Zbase_bus1=Vbase_bus1/Ibase_bus1;
135
136 %base parameters at bus 2,3,4,5
137 Vbase_net=230;
138 Ibase_net=Pbase/Vbase_net;
139 Zbase_net=Vbase_net/Ibase_net;
140
141 R12=(0.2)*Zbase_bus1/Zbase_net;          %Line resistance
      , referred to HV side
142 X12=(0.8)*Zbase_bus1/Zbase_net;          %Line inductive
      reactance, referred to HV side
143
144 R23=0.1;          %Line resistance
145 X23=0.3;          %Line inductive reactance
146
147 R24=0.1;          %Line resistance
148 X24=0.3;          %Line inductive reactance
149
```

```
150 R25=0.1;          %Line resistance
151 X25=0.3;          %Line inductive reactance
152
153
154 %Parameters in Per unit
155
156
157 %Synchronous generator is the slack bus
158 Vg(1) = Vgen_nominal/Vbase_bus1;
159 angle_g(1) = 0;
160
161 %Transmission Line 12
162 R12pu = R12/Zbase_bus1;
163 X12pu = X12/Zbase_bus1;
164 Z12pu = R12pu + 1i*X12pu;
165 G_12 = R12pu/(R12pu^2+X12pu^2);          %Per-unit
    Conductance
166 B_12 = -X12pu/(R12pu^2+X12pu^2);         %Per-unit
    Susceptance
167 Amat12 = [X12pu/abs(Z12pu) -R12pu/abs(Z12pu);R12pu/abs(
    Z12pu) X12pu/abs(Z12pu)];
168
169 %Y_bus matrix data
170 G21=-G_12;
171 B21=-B_12;
172 G12=G21;
173 B12=B21;
174
175 %Transmission Line 23
176 R23pu = R23/Zbase_net;
177 X23pu = X23/Zbase_net;
178 Z23pu = R23pu + 1i*X23pu;
179 G_23 = R23pu/(R23pu^2+X23pu^2);          %Per-unit
    Conductance
180 B_23= -X23pu/(R23pu^2+X23pu^2);         %Per-unit
    Susceptance
```

```
181 Amat23 = [X23pu/abs(Z23pu) -R23pu/abs(Z23pu);R23pu/abs(
    Z23pu) X23pu/abs(Z23pu)];
182 INVAmat23 = inv(Amat23);
183
184 %Y_bus matrix data
185 G23=-G_23;
186 B23=-B_23;
187 G32=G23;
188 B32=B23;
189
190 %Transmission Line 24
191 R24pu = R24/Zbase_net;
192 X24pu = X24/Zbase_net;
193 Z24pu = R24pu + 1i*X24pu;
194 G_24 = R24pu/(R24pu^2+X24pu^2);          %Per-unit
    Conductance
195 B_24= -X24pu/(R24pu^2+X24pu^2);          %Per-unit
    Susceptance
196 Amat24 = [X24pu/abs(Z24pu) -R24pu/abs(Z24pu);R24pu/abs(
    Z24pu) X24pu/abs(Z24pu)];
197 INVAmat24 = inv(Amat24);
198
199 %Y_bus matrix data
200 G24=-G_24;
201 B24=-B_24;
202 G42=G24;
203 B42=B24;
204
205 %Transmission Line 25
206 R25pu = R25/Zbase_net;
207 X25pu = X25/Zbase_net;
208 Z25pu = R25pu + 1i*X25pu;
209 G_25 = R25pu/(R25pu^2+X25pu^2);          %Per-unit
    Conductance
210 B_25= -X25pu/(R25pu^2+X25pu^2);          %Per-unit
    Susceptance
```

```
211 Amat25 = [X25pu/abs(Z25pu) -R25pu/abs(Z25pu);R25pu/abs(
        Z25pu) X25pu/abs(Z25pu)];
212 INVAmat25 = inv(Amat25);
213
214 %Y_bus matrix data
215 G25=-G_25;
216 B25=-B_25;
217 G52=G25;
218 B52=B25;
219
220 %Y_bus matrix data
221 G11 = G_12;
222 B11 = B_12;
223 G22 = G_12 + G_23 + G_24 + G_25;
224 B22 = B_12 + B_23 + B_24 + B_25;
225 G33 = G_23;
226 B33 = B_23;
227 G44 = G_24;
228 B44 = B_24;
229 G55 = G_25;
230 B55 = B_25;
231
232 Ymatrix = [ G11+1i*B11 G12+1i*B12 0 0 0;...
233             G21+1i*B21 G22+1i*B22 G23+1i*B23 G24+1i*B24
                    G25+1i*B25;...
234             0 G32+1i*B32 G33+1i*B33 0 0;...
235             0 G42+1i*B42 0 G44+1i*B44 0;...
236             0 G52+1i*B52 0 0 G55+1i*B55 ];
237 %Load bus 3
238 %Pload3 = Pload_bus3/Pbase;
239 Qload3 = Qload_bus3/Pbase;
240
241 %Load bus 4
242 %Pload4 = Pload_bus4/Pbase;
243 Qload4 = Qload_bus4/Pbase;
244
```

```
245 %Load bus 5
246 %Pload5 = Pload_bus5/Pbase;
247 Qload5 = Qload_bus5/Pbase;
248
249
250 %PV SYSTEM
251
252 C_pcc = 50e-6;  %PCC capacitance
253 L_pcc = 1.35e-3;    %LC filter inductance
254 R_pcc = 0.1;
255
256 %PV system 3
257 Spv_RATED3 = 10e3;    %10kVA rating inverter
258 PV_RATED3 = 8e3;      %8kVA rating solar panel
259
260 %PV system 4
261 Spv_RATED4 = 10e3;    %10kVA rating inverter
262 PV_RATED4 = 7.5e3;      %8kVA rating solar panel
263
264 %PV system 5
265 Spv_RATED5 = 10e3;    %10kVA rating inverter
266 PV_RATED5 = 7e3;      %8kVA rating solar panel
267
268 %Battery system 3
269 B3_CAPACITY(1) = 1.5*1e3*3600/Pbase;     %2kWh battery
    capacity but we will operate it at 1.5kWh
270 B3_PowerRating = 10*1e3/Pbase;      %10kW power rating
271
272 %Battery system 4
273 B4_CAPACITY(1) = 1.5*1e3*3600/Pbase;     %2kWh battery
    capacity but we will operate it at 1.5kWh
274 B4_PowerRating = 10*1e3/Pbase;      %10kW power rating
275
276 %Battery system 5
277 B5_CAPACITY(1) = 1.5*1e3*3600/Pbase;     %2kWh battery
    capacity but we will operate it at 1.5kWh
```

```
278 B5_PowerRating = 10*1e3/Pbase;      %10kW power rating
279
280
281 %PI controller gains
282 Kp_V = 0.385;
283 Ki_V = 46.3;
284 Kp_I = 17.38;
285 Ki_I = 4444.75;
286
287
288
289 %Generator parameters in Per Unit (Data from slack bus
      in Sauer's book)
290
291 fs=60;                                          %nominal
      frequency in hertz
292 ws = 2*pi*fs;                                   %nominal
      frequency in radians
293
294 %Machine data
295 H=23.64;
296 M=(2*H/ws);
297 % D=0.1*M;
298 D=0;
299 Rs=0;
300 Xd=0.146;
301 Xdp=0.0608;
302 Xq=0.0969;
303 Xqp=0.0969;
304 Tdop=8.96;
305 Tqop=0.31;
306
307 %Exciter data
308 KA=20;
309 TA=0.2;
310 KE=1;
```

```
311 TE=0.314;
312 KF=0.063;
313 TF=0.35;
314 alpha_comp=0.1;
315
316 %Speed governor data
317 TSV=0.25;
318 TCH=0.5;
319 RD=5/100;                                          %5
      percent droop
320 RD_pv=0.05/100;                                    %0.3
      percent droop on PV
321 RD_Q=0.5/100;
322
323
324 tau=5;
325 AGC_time = 10;
326 next_AGC = AGC_time;
327 xxxM=0;
328
329
330
331 %Network Load Flow (Synchronous Generator is the Slack
      bus)
332
333 for j=0:MPPdatapoints
334     MPPxxx(j+1) = ( exp((-1*((j*2)-MPPdatapoints).^2)
          /(2*(MPPdatapoints/3)^2)) );
335     %7kW solar panel, MPP changes every 2seconds
336 end
337 MPPxxx_3 = PV_RATED3*MPPxxx;
338 MPPxxx_4 = PV_RATED4*MPPxxx;
339 MPPxxx_5 = PV_RATED5*MPPxxx;
340
341 CloudCover
342
```

```
343 Pload3 = PloadX(1);
344 Pload4 = PloadX(1);
345 Pload5 = PloadX(1);
346 Pmax = 0.6;
      %max load before 4pm
347
348 Initialize_withAGC_5bus_inertia
349
350 for i=1:m-1
351
352         t = (i+1)*h; %time in seconds
353
354
355         %Synchronize PV system to network at t=0seconds
              (Same frequency as bus) as a PQ bus
356
357         %set PV reactive power supply to load reactive
              power, compute MPP
358
359         if Turn_off_SG(i) == 0
360             SynchronousGenerator    %call synchronous
                  generator dynamics after SS
361         end
362
363
364
365         %  MPP, BattMPP, Pload values at time step
366             MPP_BattMPP_Qrated_Pload_inertia
367
368
369
370         if ( (MPP3(i+1) + MPP4(i+1) + MPP5(i+1))<=(Pmax
              *3) )&&Turn_off_SG(i)==1     %if MPP is less
              than or equal to a value Pload
371
372             %AGC does this
```

```
373             Turn_off_SG(i)=0;                   %Turn ON SG
                    must occur before limit is reached
374             SynchGenInitial_withAGC2%%%%%%%%%%%%Re-
                   initialize if SG was initially off
375             SynchronousGenerator                %call
                   synchronous generator dynamics after SS
376             PC_pv3 = P3/BattMPP3(i+1);
377             PC_pv4 = P4/BattMPP4(i+1);
378             PC_pv5 = P5/BattMPP5(i+1);
379             xxxM=1;
380         end
381
382
383         if Turn_off_SG(i)==0
384
385             w_sys(i+1) = wg(i+1);
386             PVsystem2_withAGC_5bus_inertia
                                %call PV system fast
                   dynamics after SS
387             Network_Equation_SG_ONorOFF_5bus
388
389         elseif Turn_off_SG(i)==1
390
391             %PV control dynamics reach steady state in
                   one time step
392             V_pv3(i+1) = V_pv_ref3(i);
393             angle_pv3(i+1) = angle_pv3(i)+((w_pv3(i)-ws
                   )*(h)*180/pi);
394             V_pv4(i+1) = V_pv_ref4(i);
395             angle_pv4(i+1) = angle_pv4(i)+((w_pv4(i)-ws
                   )*(h)*180/pi);
396             V_pv5(i+1) = V_pv_ref5(i);
397             angle_pv5(i+1) = angle_pv5(i)+((w_pv5(i)-ws
                   )*(h)*180/pi);
398
```

```
399             %w_pv3(i+1) = w_pv3(i)+ (h/tau)*(-w_pv3(i)
                    + ws - m_pv3(i)*(-PC_pv*MPP(i) + P_pvp))
                    ;    %modelling droop control as a
                    dynamic system
400
401             Network_Equation_SG_ONorOFF_5bus     %needs
                    V_pv3(i+1)and angle_pv3(i+1) and
                    determines P_pv(i+1) and Q_pv(i+1)
402             PVsystem2_withAGC_5bus_inertia
                                %call PV system fast
                    dynamics after SS
403             w_sys(i+1) = (w_pv3(i+1)+w_pv4(i+1)+w_pv5(i
                    +1))/3;       %system frequency
404
405         end
406
407
408         Turn_off_SG(i+1) = Turn_off_SG(i);
409
410         %Every AGC_time seconds, AGC kicks in
411         if t==next_AGC
412             AGC_5bus_inertia
413             AGC_Network_Equation_SG_ONorOFF_5bus_inertia

414             next_AGC = next_AGC + AGC_time;
415         end
416
417         clc
418         toc
419 end
420
421 B3_CAPACITY = B3_CAPACITY*Pbase / (3600*1e3);        %
    Battery 3 capacity in kWh
422 B4_CAPACITY = B4_CAPACITY*Pbase / (3600*1e3);        %
    Battery 4 capacity in kWh
```

```
423 B5_CAPACITY = B5_CAPACITY*Pbase / (3600*1e3);          %
    Battery 5 capacity in kWh
```

## B.2 Initial values

### B.2.1 Initial Power and Frequency

```
1  P_pv3(1) = MPP3(1);
2  P_PVDC3(1) = P_pv3(1);
3
4  P_pv4(1) = MPP4(1);
5  P_PVDC4(1) = P_pv4(1);
6
7  P_pv5(1) = MPP5(1);
8  P_PVDC5(1) = P_pv5(1);
9
10 w = ws;
11
12 Px3 = Pload3 - P_pv3(1);
13 Px4 = Pload4 - P_pv4(1);
14 Px5 = Pload5 - P_pv5(1);
```

### B.2.2 PV Parameters Initial Values

```
1  global  i
2
3
4  %PV system INITIAL VALUES
5
6
7  PC_pvQ = RD_Q*(Q_pv3(i)/Qratedp3);
8
9  PC_pv3 = 1;
10
11 PC_pv4 = 1;
```

```
12
13 PC_pv5 = 1;
14
15 m_pv3(i) = (RD*ws)/(MPP3(i));                %5percent
       frequency droop
16 m_pv23(i) = (RD_pv*ws)/(MPP3(i));              %0.3
       percent frequency droop
17 w_pv3(i) = ws - (m_pv3(i)*P_pv3(i))+(RD*ws*PC_pv3);
18
19 m_pv4(i) = (RD*ws)/(MPP4(i));                %5percent
       frequency droop
20 m_pv24(i) = (RD_pv*ws)/(MPP4(i));              %0.3
       percent frequency droop
21 w_pv4(i) = ws - (m_pv4(i)*P_pv4(i))+(RD*ws*PC_pv4);
22
23 m_pv5(i) = (RD*ws)/(MPP5(i));                %5percent
       frequency droop
24 m_pv25(i) = (RD_pv*ws)/(MPP5(i));              %0.3
       percent frequency droop
25 w_pv5(i) = ws - (m_pv5(i)*P_pv5(i))+(RD*ws*PC_pv5);
26
27
28 V_pv_ref3(i) = V_pv3(i);
29 V_pv_ref4(i) = V_pv4(i);
30 V_pv_ref5(i) = V_pv5(i);
```

### B.2.3 Microturbine Parameters Initial Values

```
1 global  i
2
3 %GENERATOR INITIAL VALUES
4
5
6 Ig_network = (Pg(i)-1i*Qg(i))/(Vg(i)*exp(-1i*angle_g(i)
       *pi/180));
```

```
%Synchronous generator current in synchronous reference frame and network
...base

Ig = Ig_network*Ibase_bus1/Ibase_gen;
I_g(i)=abs(Ig);
%Synchronous generator current in synchronous reference frame and generator
...base

delta(i)= 180*(angle(Vg(i)*exp(1i*angle_g(i)*pi/180)+(Rs+(1i*Xq))*Ig))/pi;
%Rotor position in degrees (STEP 2, PAGE 187)

Idq_G = Ig*exp(-1i*(delta(i)-90)*pi/180);
%Synchronous generator current in rotor reference frame

Id_G = real(Idq_G);

Iq_G = imag(Idq_G);

Vd_q = Vg(i)*exp(1i*(angle_g(i)-delta(i)+90)*pi/180);
%Synchronous generator bus voltage in rotor reference frame

Vd_G = real(Vd_q);

Vq_G = imag(Vd_q);

Edp(i) = (Xq-Xqp)*Iq_G;

Eqp(i) = Vq_G + Rs*Iq_G + Xdp*Id_G;

Efd(i) = Eqp(i)+(Xd-Xdp)*Id_G;

VR(i) = (KE+0.0039*exp(1.555*Efd(i)))*Efd(i);
```

```
39
40 Rf(i) = (KF/TF)*Efd(i);
41
42 Vref = Vg(i)+VR(i)/KA;
43
44 TM(i) = Edp(i)*Id_G+Eqp(i)*Iq_G+(Xqp-Xdp)*Id_G*Iq_G+D*(
       w-ws);
45
46 PSV(i) = TM(i);
47
48 PC = PSV(i); % AGC control
49
50 wg(i) = ws - ws*RD*(PSV(i)-PC);
```

### B.2.4 System Initial Values

```
1  global i
2
3      i=1;
4
5      x0 = [1;0;0;Qload3;0;Qload4;0;Qload5];   %[V2;angle2
           ;angle_pv3;Q_pv3;angle_pv4;Q_pv4;angle_pv5;Q_pv5
           ]
6
7      MPP3(i) = MPPxxx_3(i)/Pbase;
8      BattMPP3(i) = MPP3(i);
9      MPP4(i) = MPPxxx_4(i)/Pbase;
10     BattMPP4(i) = MPP4(i);
11     MPP5(i) = MPPxxx_5(i)/Pbase;
12     BattMPP5(i) = MPP5(i);
13
14     Qrated3(i) = sqrt(1^2 - MPP3(i)^2);
15     Qrated4(i) = sqrt(1^2 - MPP4(i)^2);
16     Qrated5(i) = sqrt(1^2 - MPP5(i)^2);
17
```

```
18      MPPp3 = ( (X23pu*MPP3(i))-(R23pu*Qrated3(i)) )/abs(
            Z23pu);
19      MPPp4 = ( (X24pu*MPP4(i))-(R24pu*Qrated4(i)) )/abs(
            Z24pu);
20      MPPp5 = ( (X25pu*MPP5(i))-(R25pu*Qrated5(i)) )/abs(
            Z25pu);
21
22      Qratedp3 = ( (R23pu*MPP3(i))+(X23pu*Qrated3(i)) )/
            abs(Z23pu);
23      Qratedp4 = ( (R24pu*MPP4(i))+(X24pu*Qrated4(i)) )/
            abs(Z24pu);
24      Qratedp5 = ( (R25pu*MPP5(i))+(X25pu*Qrated5(i)) )/
            abs(Z25pu);
25
26      n_pv3(i) = (RD_Q)/(Qratedp3);                %2
            percent voltage droop
27      n_pv4(i) = (RD_Q)/(Qratedp4);                %2
            percent voltage droop
28      n_pv5(i) = (RD_Q)/(Qratedp5);                %2
            percent voltage droop
29
30      %Initial conditions of power demand on PV system
31
32      Initial_Power_and_Frequency_AGC_5bus_inertia
33
34      options = optimset('Display','iter','TolFun',1e-12,
            'TolX',1e-12);   % Option to display output
35      [x,fval1] = fsolve(@NetworkLoadFlow_5bus,x0,options
            ); % Call solver  keep V_pv(i) at 1pu
36      %[V2;angle2;angle_pv3;Q_pv3;angle_pv4;Q_pv4;
            angle_pv5;Q_pv5]
37
38      V2(i) = x(1);
39      V_pv3(i) = 1;
40      V_pv4(i) = 1;
41      V_pv5(i) = 1;
```

```
42      angle2(i) = x(2);
43      angle_pv3(i) = x(3);
44      angle_pv4(i) = x(5);
45      angle_pv5(i) = x(7);
46
47      % Real and reactive generation at the slack bus:
48      Pg(i) = G11*Vg(i)^2 + Vg(i)*V2(i)*(G12*(cosd(
            angle_g(i)-angle2(i))) ...
49              + B12*(sind(angle_g(i)-angle2(i))));
50
51      Qg(i) = -B11*Vg(i)^2 + Vg(i)*V2(i)*(G12*(sind(
            angle_g(i)-angle2(i))) ...
52              - B12*(cosd(angle_g(i)-angle2(i))));
53
54      Q_pv3(i) = x(4);
55      Q_pv4(i) = x(6);
56      Q_pv5(i) = x(8);
57
58      PpQp3 = Amat23*[P_pv3(i);Q_pv3(i)];
59      P_pvp3 = PpQp3(1);
60      Q_pvp3 = PpQp3(2);
61
62      PpQp4 = Amat24*[P_pv4(i);Q_pv4(i)];
63      P_pvp4 = PpQp4(1);
64      Q_pvp4 = PpQp4(2);
65
66      PpQp5 = Amat25*[P_pv5(i);Q_pv5(i)];
67      P_pvp5 = PpQp5(1);
68      Q_pvp5 = PpQp5(2);
69
70      SynchGenInitial_withAGC   %call synchronous
            generator initial conditions
71
72
73      PVsystem_INITIAL_5bus
```

### B.2.5 Network Equations

```
function F = NetworkLoadFlow_5bus(x)
%Load flow equations to determine initial conditions for a 2-bus network

global  Vg angle_g Pload3 Qload3 Pload4 Qload4 Pload5 Qload5...
        G22 B22 G33 B33 G44 B44 G55 B55 G21 B21 G23 G32 B23 B32 G24 G42 B24 B42 G25 G52 B25 B52...
        i P_pv3 P_pv4 P_pv5...


F= [-P_pv3(i) + Pload3 + G33*1*1 + 1*x(1)*(G32*(cosd(x(3)-x(2))) + B32*(sind(x(3)-x(2))));...
    -P_pv4(i) + Pload4 + G44*1*1 + 1*x(1)*(G42*(cosd(x(5)-x(2))) + B42*(sind(x(5)-x(2))));...
    -P_pv5(i) + Pload5 + G55*1*1 + 1*x(1)*(G52*(cosd(x(7)-x(2))) + B52*(sind(x(7)-x(2))));...
    G22*x(1)*x(1) + x(1)*Vg(i)*(G21*(cosd(x(2)-angle_g(i))) + B21*(sind(x(2)-angle_g(i))))...
                  + x(1)*1*(G23*(cosd(x(2)-x(3))) + B23*(sind(x(2)-x(3))))...
                  + x(1)*1*(G24*(cosd(x(2)-x(5))) + B24*(sind(x(2)-x(5))))...
                  + x(1)*1*(G25*(cosd(x(2)-x(7))) + B25*(sind(x(2)-x(7))));...
    -x(4) + Qload3 - B33*1*1 + 1*x(1)*(G32*(sind(x(3)-x(2))) - B32*(cosd(x(3)-x(2))));...
    -x(6) + Qload4 - B44*1*1 + 1*x(1)*(G42*(sind(x(5)-x(2))) - B42*(cosd(x(5)-x(2))));...
    -x(8) + Qload5 - B55*1*1 + 1*x(1)*(G52*(sind(x(7)-x(2))) - B52*(cosd(x(7)-x(2))));...
    -B22*x(1)*x(1) + x(1)*Vg(i)*(G21*(sind(x(2)-angle_g(i))) - B21*(cosd(x(2)-angle_g(i))))...
```

```
20                          + x(1)*1*(G23*(sind(x(2)-x(3))) -
                                B23*(cosd(x(2)-x(3)))) ...
21                          + x(1)*1*(G24*(sind(x(2)-x(5))) -
                                B24*(cosd(x(2)-x(5)))) ...
22                          + x(1)*1*(G25*(sind(x(2)-x(7))) -
                                B25*(cosd(x(2)-x(7)))) ...
23      ];
24  end
```

## B.3 PV generator

### B.3.1 PV system

```
1  global i
2
3  %INSTANTANEOUS
4
5
6  %Photovoltaic system for AJ Microgrid
7
8  %P_pv = Output from PV system to microgrid
9  %P_PVDC = Solarpanel output
10 %P_PVDC - P_pv = Charging rate of battery in kW
11
12
13
14 %Energy gained by battery during this last time step
15 E_stored3 = (P_PVDC3(i) - P_pv3(i))*h;
16 E_stored4 = (P_PVDC4(i) - P_pv4(i))*h;
17 E_stored5 = (P_PVDC5(i) - P_pv5(i))*h;
18
19 B3_CAPACITY(i+1) = B3_CAPACITY(i) + E_stored3;
20 B4_CAPACITY(i+1) = B4_CAPACITY(i) + E_stored4;
21 B5_CAPACITY(i+1) = B5_CAPACITY(i) + E_stored5;
22
```

```
23
24 %Droop coefficients
25 m_pv3(i+1) = (RD*ws)/(BattMPP3(i+1));              %5
       percent frequency droop
26 m_pv23(i+1) = (RD_pv*ws)/(BattMPPp3);             %0.3
       percent frequency droop
27 n_pv3(i+1) = (RD_Q*1)/Qratedp3;                   %2
       percent voltage droop
28
29
30 m_pv4(i+1) = (RD*ws)/(BattMPP4(i+1));              %5
       percent frequency droop
31 m_pv24(i+1) = (RD_pv*ws)/(BattMPPp4);             %0.3
       percent frequency droop
32 n_pv4(i+1) = (RD_Q*1)/Qratedp4;                   %2
       percent voltage droop
33
34
35 m_pv5(i+1) = (RD*ws)/(BattMPP5(i+1));              %5
       percent frequency droop
36 m_pv25(i+1) = (RD_pv*ws)/(BattMPPp5);             %0.3
       percent frequency droop
37 n_pv5(i+1) = (RD_Q*1)/Qratedp5;                   %2
       percent voltage droop
38
39
40 %SG ON
41 if Turn_off_SG(i) == 0
42
43          w_pv3(i+1) = ( wg(i+1) + w_pv3(i) )/2;
44          Vpv3 = 1 - n_pv3(i+1)*(-PC_pvQ*Qratedp3 +
               Q_pvp3);                    %reference RMS
               value of bus voltage
45          V_pv_ref3(i+1) = ( Vpv3+V_pv3(i) )/2;
46          Q_pvp3 = ( 1 - V_pv_ref3(i+1) + (RD_Q*PC_pvQ))/
               n_pv3(i+1);          %Q_pv' = (R12pu*P_pv +
```

```
    X12pu*Q_pv)/abs(Z12pu)
47
48
49          w_pv4(i+1) = ( wg(i+1) + w_pv4(i) )/2;
50          Vpv4 = 1 - n_pv4(i+1)*(-PC_pvQ*Qratedp4 +
                Q_pvp4);                    %reference RMS
                value of bus voltage
51          V_pv_ref4(i+1) = ( Vpv4+V_pv4(i) )/2;
52          Q_pvp4 = ( 1 - V_pv_ref4(i+1) + (RD_Q*PC_pvQ))/
                n_pv4(i+1);             %Q_pv' = (R12pu*P_pv +
                X12pu*Q_pv)/abs(Z12pu)
53
54
55          w_pv5(i+1) = ( wg(i+1) + w_pv5(i) )/2;
56          Vpv5 = 1 - n_pv5(i+1)*(-PC_pvQ*Qratedp5 +
                Q_pvp5);                    %reference RMS
                value of bus voltage
57          V_pv_ref5(i+1) = ( Vpv5+V_pv5(i) )/2;
58          Q_pvp5 = ( 1 - V_pv_ref5(i+1) + (RD_Q*PC_pvQ))/
                n_pv5(i+1);             %Q_pv' = (R12pu*P_pv +
                X12pu*Q_pv)/abs(Z12pu)
59
60
61          %Droop control kicks in
62
63          P_pvp3 = (ws - w_pv3(i+1) + (RD*ws*PC_pv3))/
                m_pv3(i+1);               %P_pv' = (X12pu*P_pv
                 - R12pu*Q_pv)/abs(Z12pu)
64          P_pv3(i+1) = P_pvp3;

              %since frequency is dependent on P not P'
              %%% Power output from P_pv3 to microgrid
65          P_PVDC3(i+1) = P_pv3(i+1);
                                                          %
                Solar panel output controlled
                instantaneously to match PV system output
```

```
66          %quick response to frequency change is
               advantageous to microgrid frequency
               stabilization
67
68          P_pvp4 = (ws - w_pv4(i+1) + (RD*ws*PC_pv4))/
               m_pv4(i+1);                %P_pv' = (X12pu*P_pv
                - R12pu*Q_pv)/abs(Z12pu)
69          P_pv4(i+1) = P_pvp4;

               %since frequency is dependent on P not P'
               %%% Power output from P_pv4 to microgrid
70          P_PVDC4(i+1) = P_pv4(i+1);
                                                          %
               Solar panel output controlled
               instantaneously to match PV system output
71          %quick response to frequency change is
               advantageous to microgrid frequency
               stabilization
72
73          P_pvp5 = (ws - w_pv5(i+1) + (RD*ws*PC_pv5))/
               m_pv5(i+1);                %P_pv' = (X12pu*P_pv
                - R12pu*Q_pv)/abs(Z12pu)
74          P_pv5(i+1) = P_pvp5;

               %since frequency is dependent on P not P'
               %%% Power output from P_pv5 to microgrid
75          P_PVDC5(i+1) = P_pv5(i+1);
                                                          %
               Solar panel output controlled
               instantaneously to match PV system output
76          %quick response to frequency change is
               advantageous to microgrid frequency
               stabilization
77
78  %          %solve for real and reactive power
79  %          Bmat12 = [P_pvp;Q_pvp];
```

```
80 %
81 %           X = linsolve(Amat12,Bmat12);
82 %
83 %           P_pv(i+1) = X(1);
84 %           Qpv = X(2);
85
86          Qpv3 = ( Q_pvp3 - (Amat23(2,1)*P_pv3(i+1)) )/
                Amat23(2,2);
87          Qpv4 = ( Q_pvp4 - (Amat24(2,1)*P_pv4(i+1)) )/
                Amat24(2,2);
88          Qpv5 = ( Q_pvp5 - (Amat25(2,1)*P_pv5(i+1)) )/
                Amat25(2,2);
89
90          %constraints on P_pv(i+1) and P_PVDC(i+1)
91
92          %PV 3
93                  if P_pvp3>max(BattMPP3(i+1),MPP3(i+1))
                                                %Power
                    reference greater than MPP, limit
                    power reference and PV DC output
94                      P_pv3(i+1) = max(BattMPP3(i+1),MPP3
                            (i+1));
95                  end
96
97                  if P_pvp3<0                     %zero
                    limiter
98                      P_pv3(i+1)=0;
99                      P_PVDC3(i+1)=P_pv3(i+1);
100                 end
101
102         %PV 4
103                 if P_pvp4>max(BattMPP4(i+1),MPP4(i+1))
                                                %Power
                    reference greater than MPP, limit
                    power reference and PV DC output
```

```
104                     P_pv4(i+1) = max(BattMPP4(i+1),MPP4
                            (i+1));
105                 end
106
107                 if P_pvp4<0                         %zero
                        limiter
108                     P_pv4(i+1)=0;
109                     P_PVDC4(i+1)=P_pv4(i+1);
110                 end
111
112             %PV 5
113                 if P_pvp5>max(BattMPP5(i+1),MPP5(i+1))
                                                %Power
                        reference greater than MPP, limit
                        power reference and PV DC output
114                     P_pv5(i+1) = max(BattMPP5(i+1),MPP5
                            (i+1));
115                 end
116
117                 if P_pvp5<0                         %zero
                        limiter
118                     P_pv5(i+1)=0;
119                     P_PVDC5(i+1)=P_pv5(i+1);
120                 end
121
122
123             %constraint on Q_pvp
124                 if Qpv3>Qrated3(i+1)
125                     Qpv3=Qratedp3;
126                 elseif Qpv3<0
127                     Qpv3=0;
128                 end
129
130                 if Qpv4>Qrated4(i+1)
131                     Qpv4=Qratedp4;
132                 elseif Qpv4<0
```

```
133                     Qpv4=0;
134                 end
135
136                 if Qpv5>Qrated5(i+1)
137                     Qpv5=Qratedp5;
138                 elseif Qpv5<0
139                     Qpv5=0;
140                 end
141
142
143                     %check battery state and charge
                            if necessary%
144                             PV_DC_output_inertia
145
146
147
148         Q_pv3(i+1) = Q_pv3(i) + h*(-Q_pv3(i) + Qpv3)/8;
                                    %Reactive power output
            with delay
149         Q_pv4(i+1) = Q_pv4(i) + h*(-Q_pv4(i) + Qpv4)/8;
                                    %Reactive power output
            with delay
150         Q_pv5(i+1) = Q_pv5(i) + h*(-Q_pv5(i) + Qpv5)/8;
                                    %Reactive power output
            with delay
151
152
153         X = Amat23*[P_pv3(i+1);Q_pv3(i+1)];
154         Q_pvp3 = X(2);
155         X = Amat24*[P_pv4(i+1);Q_pv4(i+1)];
156         Q_pvp4 = X(2);
157         X = Amat25*[P_pv5(i+1);Q_pv5(i+1)];
158         Q_pvp5 = X(2);
159 end
160
161
```

```
%SG OFF
if Turn_off_SG(i) == 1


    %check battery state and charge if necessary%
                PV_DC_output_inertia



    X = Amat23*[P_pv3(i+1);Q_pv3(i+1)];
    P_pvp3 = X(1);
    Q_pvp3 = X(2);
    X = Amat24*[P_pv4(i+1);Q_pv4(i+1)];
    P_pvp4 = X(1);
    Q_pvp4 = X(2);
    X = Amat25*[P_pv5(i+1);Q_pv5(i+1)];
    P_pvp5 = X(1);
    Q_pvp5 = X(2);

    w_pv3(i+1) = ws - m_pv23(i+1)*(-PC_pv3*BattMPPp3 +
        P_pvp3);
    w_pv4(i+1) = ws - m_pv24(i+1)*(-PC_pv4*BattMPPp4 +
        P_pvp4);
    w_pv5(i+1) = ws - m_pv25(i+1)*(-PC_pv5*BattMPPp5 +
        P_pvp5);

    V_pv_ref3(i+1) = 1 - n_pv3(i+1)*(-PC_pvQ*Qratedp3 +
        Q_pvp3);            %reference RMS value of bus
        voltage
    V_pv_ref4(i+1) = 1 - n_pv4(i+1)*(-PC_pvQ*Qratedp4 +
        Q_pvp4);            %reference RMS value of bus
        voltage
    V_pv_ref5(i+1) = 1 - n_pv5(i+1)*(-PC_pvQ*Qratedp5 +
        Q_pvp5);            %reference RMS value of bus
        voltage

```

```
189 end
```

### B.3.2 Power Function

```
1  global i
2  %
3          j = floor(i/MPPintervals);
4          jj = floor(i/SlowCloudInterval);
5          MPP3(i+1) = MPPxxx_3(j+1)/Pbase ;
6          SlowCloudState(i+1) =  CloudCoverStateSLOW(jj
               +1);
7
8              if SlowCloudState(i+1)<SlowCloudState(i)
9
10                 SlowCloudState(i+1) = SlowCloudState(i)
                       - SlowPerTimeStep;
11
12             elseif SlowCloudState(i+1)>SlowCloudState(i
                   )
13
14                     SlowCloudState(i+1) =
                           SlowCloudState(i) +
                           SlowPerTimeStep;
15
16             elseif SlowCloudState(i+1)==SlowCloudState(
                   i)
17
18                 SlowCloudState(i+1) = SlowCloudState(i)
                       ;
19
20             end
21
22
23 %          MPP3(i+1) = MPP3(i+1)*(1+CloudCoverState(i+1)
       +SlowCloudState(i+1));            %CloudCover
24
```

```
25          %MPP3(i+1) = PV_RATED3/Pbase;
                                          %Test frequency
                change
26          rate3 = abs(((MPP3(i+1) - MPP3(i))*Pbase)/h);
                        %dMPP/dt in W/s
27              if rate3 > 40
28                  rate3=40;
29              end
30
31              if BattMPP3(i)<MPP3(i+1)
32                  BattMPP3(i+1) = BattMPP3(i)+((rate3/
                        Pbase)*h);
33              elseif BattMPP3(i)>MPP3(i+1)
34                  BattMPP3(i+1) = BattMPP3(i)-((rate3/
                        Pbase)*h);
35              elseif BattMPP3(i)==MPP3(i+1)
36                  BattMPP3(i+1) = BattMPP3(i);
37              end
38          Qrated3(i+1) = sqrt(1^2 - BattMPP3(i+1)^2);
39          BattMPPp3 = ( (X23pu*BattMPP3(i+1))-(R23pu*
                Qrated3(i+1)) )/abs(Z23pu);
40          Qratedp3 = ( (R23pu*BattMPP3(i+1))+(X23pu*
                Qrated3(i+1)) )/abs(Z23pu);
41
42          %
43
44          MPP4(i+1) = MPPxxx_4(j+1)/Pbase;
45
46  %           MPP4(i+1) = MPP4(i+1)*(1+CloudCoverState(i+1)
        +SlowCloudState(i+1));          %CloudCover
47
48          %MPP4(i+1) = PV_RATED4/Pbase;
                                          %Test frequency
                change
49          rate4 = abs(((MPP4(i+1) - MPP4(i))*Pbase)/h);
                        %dMPP/dt in W/s
```

```
50              if rate4 > 40
51                  rate4=40;
52              end
53
54              if BattMPP4(i)<MPP4(i+1)
55                  BattMPP4(i+1) = BattMPP4(i)+((rate4/
                        Pbase)*h);
56              elseif BattMPP4(i)>MPP4(i+1)
57                  BattMPP4(i+1) = BattMPP4(i)-((rate4/
                        Pbase)*h);
58              elseif BattMPP4(i)==MPP4(i+1)
59                  BattMPP4(i+1) = BattMPP4(i);
60              end
61          Qrated4(i+1) = sqrt(1^2 - BattMPP4(i+1)^2);
62          BattMPPp4 = ( (X24pu*BattMPP4(i+1))-(R24pu*
                Qrated4(i+1)) )/abs(Z24pu);
63          Qratedp4 = ( (R24pu*BattMPP4(i+1))+(X24pu*
                Qrated4(i+1)) )/abs(Z24pu);
64
65          %
66
67          MPP5(i+1) = MPPxxx_5(j+1)/Pbase;
68
69  %           MPP5(i+1) = MPP5(i+1)*(1+CloudCoverState(i+1)
    +SlowCloudState(i+1));              %CloudCover
70
71          %MPP5(i+1) = PV_RATED5/Pbase;
                                              %Test frequency
                change
72          rate5 = abs(((MPP5(i+1) - MPP5(i))*Pbase)/h);
                        %dMPP/dt in W/s
73              if rate5 > 40
74                  rate5=40;
75              end
76
77              if BattMPP5(i)<MPP5(i+1)
```

```
78                  BattMPP5(i+1) = BattMPP5(i)+((rate5/
                        Pbase)*h);
79              elseif BattMPP5(i)>MPP5(i+1)
80                  BattMPP5(i+1) = BattMPP5(i)-((rate5/
                        Pbase)*h);
81              elseif BattMPP5(i)==MPP5(i+1)
82                  BattMPP5(i+1) = BattMPP5(i);
83              end
84          Qrated5(i+1) = sqrt(1^2 - BattMPP5(i+1)^2);
85          BattMPPp5 = ( (X25pu*BattMPP5(i+1))-(R25pu*
                Qrated5(i+1)) )/abs(Z25pu);
86          Qratedp5 = ( (R25pu*BattMPP5(i+1))+(X25pu*
                Qrated5(i+1)) )/abs(Z25pu);
                                %BattMPP is the maximum
                Real power output of the PV system
            %

88
89          Pload3 = PloadX(i+1);
90          Pload4 = PloadX(i+1);
91          Pload5 = PloadX(i+1);
```

### B.3.3 PV Output Constraints

```
1  global i
2
3      if P_pv3(i+1)<MPP3(i+1)
4          P_PVDC3(i+1) = P_pv3(i+1);
                                                    %PV
              DC output matches PV system output to
              microgrid, so battery does not discharge
5      else
6          P_PVDC3(i+1) = MPP3(i+1);
7      end
8
9      if ( P_PVDC3(i+1)<MPP3(i+1) )&&( B3_CAPACITY(i+1)<
           B3_CAPACITY(1) )      %If possible, charge
```

```
    battery if it is not fully charged
10          P_PVDC3(i+1) = MPP3(i+1);
11      end
12
13
14
15      if P_pv4(i+1)<MPP4(i+1)
16          P_PVDC4(i+1) = P_pv4(i+1);
                                                        %PV
            DC output matches PV system output to
            microgrid, so battery does not discharge
17      else
18          P_PVDC4(i+1) = MPP4(i+1);
19      end
20
21      if ( P_PVDC4(i+1)<MPP4(i+1) )&&( B4_CAPACITY(i+1)<
        B4_CAPACITY(1) )      %If possible, charge
        battery if it is not fully charged
22          P_PVDC4(i+1) = MPP4(i+1);
23      end
24
25
26
27      if P_pv5(i+1)<MPP5(i+1)
28          P_PVDC5(i+1) = P_pv5(i+1);
                                                        %PV
            DC output matches PV system output to
            microgrid, so battery does not discharge
29      else
30          P_PVDC5(i+1) = MPP5(i+1);
31      end
32
33      if ( P_PVDC5(i+1)<MPP5(i+1) )&&( B5_CAPACITY(i+1)<
        B5_CAPACITY(1) )      %If possible, charge
        battery if it is not fully charged
34          P_PVDC5(i+1) = MPP5(i+1);
```

```
35      end
```

## B.4 Network

### B.4.1 Network Power Flow

```
1  global i
2
3
4  %NETWORK ALGEBRAIC EQUATION
5
6      if Turn_off_SG(i) == 0
7
8          x0 = [Vg(i);angle_g(i);Id_G;Iq_G;V2(i);angle2(i
               );V_pv3(i);angle_pv3(i);V_pv4(i);angle_pv4(i
               );V_pv5(i);angle_pv5(i)];
9          options = optimset('Display','iter','TolFun',1e
               -9,'TolX',1e-9);  %Option to display output
10         [x,fval2] = fsolve(@AlgebraicEquations_5bus,x0,
               options); % Call solver
11
12         Vg(i+1) = x(1);
13         angle_g(i+1) = x(2);
14         Id_G = x(3); %in synchronous generator base
15         Iq_G = x(4); %in synchronous generator base
16
17         Idq_G = Id_G + (1i*Iq_G);
18         Ig = Idq_G*exp(1i*(delta(i)-90)*pi/180);
19         I_g(i+1)=abs(Ig);
20
21         V2(i+1) = x(5);
22         angle2(i+1) = x(6);
23
24         V_pv3(i+1) = x(7);
25         angle_pv3(i+1) = x(8);
```

```
26          V_pv4(i+1) = x(9);
27          angle_pv4(i+1) = x(10);
28          V_pv5(i+1) = x(11);
29          angle_pv5(i+1) = x(12);
30
31          % Real and reactive generation at the slack bus
                :
32          Pg(i+1) = G11*Vg(i+1)^2 + Vg(i+1)*V2(i+1)*(G21
                *(cosd(angle_g(i+1)-angle2(i+1))) ...
33                  + B21*(sind(angle_g(i+1)-angle2(i+1))))
                        ;
34
35          Qg(i+1)=  -B11*Vg(i+1)^2 + Vg(i+1)*V2(i+1)*(G21
                *(sind(angle_g(i+1)-angle2(i+1))) ...
36                  - B21*(cosd(angle_g(i+1)-angle2(i+1))))
                        ;
37
38      end
39
40
41      if Turn_off_SG(i) == 1
42          x01 = [Vg(i);angle_g(i);V2(i);angle2(i);P_pv3(i
                );Q_pv3(i);P_pv4(i);Q_pv4(i);P_pv5(i);Q_pv5(
                i)];
43          options = optimset('Display','iter','TolFun',1e
                -9,'TolX',1e-9);  %Option to display output
44          [x,fval2] = fsolve(@AlgebraicEquationsNOSG_5bus
                ,x01,options); % Call solver
45
46          Vg(i+1) = x(1);
47          angle_g(i+1) = x(2);
48          V2(i+1) = x(3);
49          angle2(i+1) = x(4);
50          P_pv3(i+1) = x(5);
51          Q_pv3(i+1) = x(6);
52          P_pv4(i+1) = x(7);
```

```
53          Q_pv4(i+1) = x(8);
54          P_pv5(i+1) = x(9);
55          Q_pv5(i+1) = x(10);
56
57      end
```

### B.4.2 Network Equations

Microturbine ON

```
1  function F = AlgebraicEquations_5bus(x)
2
3  global Edp delta Rs Xdp Xqp Eqp i...
4         Ibase_bus1 Ibase_gen...
5         G11 B11 G22 B22 G33 B33 G44 B44 G55 B55 G12 B12
             G21 B21 G23 B23 G32 B32 G24 B24 G42 B42 G25
             B25 G52 B52...
6         P_pv3 Q_pv3 P_pv4 Q_pv4 P_pv5 Q_pv5 Pload3
             Qload3 Pload4 Qload4 Pload5 Qload5
7
8
9  %At voltage magnitude and phase set by PV droop control
       scheme, solve
10 %network equations for voltage and currents at
       synchronous generator
11 % [Vg(i);angle_g(i);Id_G;Iq_G;V2(i);angle2(i);V_pv3(i);
       angle_pv3(i);V_pv4(i);angle_pv4(i);V_pv5(i);
       angle_pv5(i)];
12 F = [    Edp(i+1)-x(1)*sind(delta(i+1)-x(2))- Rs*x(3) +
       Xqp*x(4);
13          Eqp(i+1)-x(1)*cosd(delta(i+1)-x(2))- Rs*x(4) -
               Xdp*x(3);
14          (Ibase_gen/Ibase_bus1)*(x(1)*x(3)*sind(delta(i
               +1)-x(2))+x(1)*x(4)*cosd(delta(i+1)-x(2)))-(
                G11*x(1)^2 + x(1)*x(5)*(G12*(cosd(x(2)-x(6)
               ))+B12*(sind(x(2)-x(6)))) );
```

```
15  (Ibase_gen/Ibase_bus1)*(x(1)*x(3)*cosd(delta(i
        +1)-x(2))-x(1)*x(4)*sind(delta(i+1)-x(2)))-(
         -B11*x(1)^2 + x(1)*x(5)*(G12*(sind(x(2)-x
        (6)))-B12*(cosd(x(2)-x(6)))) );
16  -P_pv3(i+1) + Pload3 + G33*x(7)*x(7) + x(7)*x
        (5)*(G32*(cosd(x(8)-x(6))) + B32*(sind(x(8)-
        x(6))));...
17  -Q_pv3(i+1) + Qload3 - B33*x(7)*x(7) + x(7)*x
        (5)*(G32*(sind(x(8)-x(6))) - B32*(cosd(x(8)-
        x(6))));...
18  -P_pv4(i+1) + Pload4 + G44*x(9)*x(9) + x(9)*x
        (5)*(G42*(cosd(x(10)-x(6))) + B42*(sind(x
        (10)-x(6))));...
19  -Q_pv4(i+1) + Qload4 - B44*x(9)*x(9) + x(9)*x
        (5)*(G42*(sind(x(10)-x(6))) - B42*(cosd(x
        (10)-x(6))));...
20  -P_pv5(i+1) + Pload5 + G55*x(11)*x(11) + x(11)*
        x(5)*(G52*(cosd(x(12)-x(6))) + B52*(sind(x
        (12)-x(6))));...
21  -Q_pv5(i+1) + Qload5 - B55*x(11)*x(11) + x(11)*
        x(5)*(G52*(sind(x(12)-x(6))) - B52*(cosd(x
        (12)-x(6))));...
22  G22*x(5)*x(5) + x(5)*x(1)*(G21*(cosd(x(6)-x(2))
        ) + B21*(sind(x(6)-x(2))))...
23                  + x(5)*x(7)*(G23*(cosd(x(6)-x(8))
                        ) + B23*(sind(x(6)-x(8))))...
24                  + x(5)*x(9)*(G24*(cosd(x(6)-x(10)
                        )) + B24*(sind(x(6)-x(10))))
                        ...
25                  + x(5)*x(11)*(G25*(cosd(x(6)-x
                        (12))) + B25*(sind(x(6)-x(12))
                        ));...
26  -B22*x(5)*x(5) + x(5)*x(1)*(G21*(sind(x(6)-x(2)
        )) - B21*(cosd(x(6)-x(2))))...
27                    + x(5)*x(7)*(G23*(sind(x(6)-x(8)
                          )) - B23*(cosd(x(6)-x(8))))
```

```
                                    ...
28                                + x(5)*x(9)*(G24*(sind(x(6)-x
                                      (10))) - B24*(cosd(x(6)-x(10)
                                      ))) ...
29                                + x(5)*x(11)*(G25*(sind(x(6)-x
                                      (12))) - B25*(cosd(x(6)-x(12)
                                      ))) ...
30      ];
31  end
32    %(7.13) in Sauer's book
33    %(7.14) in Sauer's book
34    %(7.28) in Sauer's book
35    %(7.29) in Sauer's book
```

Microturbine OFF

```
1   function F = AlgebraicEquationsNOSG_5bus(x)
2
3   global i  G11 B11 G22 B22 G33 B33 G44 B44 G55 B55 G12
      B12 G21 B21 G23 B23 G32 B32 G24 B24 G42 B42 G25 B25
      G52 B52...
4           Pg Qg Pload3 Qload3 Pload4 Qload4 Pload5 Qload5
                ...
5           V_pv3 angle_pv3 V_pv4 angle_pv4 V_pv5 angle_pv5
6
7   %At voltage magnitude and phase set by PV droop control
       scheme, solve
8   %network equations for voltage and currents at
      synchronous generator
9   %    [Vg(i);angle_g(i);V2(i);angle2(i);P_pv3(i);Q_pv3(i)
      ;P_pv4(i);Q_pv4(i);P_pv5(i);Q_pv5(i)]
10  F = [     Pg(i+1)-( G11*x(1)^2 + x(1)*x(3)*(G12*(cosd(x
      (2)-x(4)))+B12*(sind(x(2)-x(4)))) );
11          Qg(i+1)-( -B11*x(1)^2 + x(1)*x(3)*(G12*(sind(x
                (2)-x(4)))-B12*(cosd(x(2)-x(4)))) );
```

```
12          -x(5) + Pload3 + G33*V_pv3(i+1)*V_pv3(i+1) +
               V_pv3(i+1)*x(3)*(G32*(cosd(angle_pv3(i+1)-x
               (4))) + B32*(sind(angle_pv3(i+1)-x(4))));
13          -x(6) + Qload3 - B33*V_pv3(i+1)*V_pv3(i+1) +
               V_pv3(i+1)*x(3)*(G32*(sind(angle_pv3(i+1)-x
               (4))) - B32*(cosd(angle_pv3(i+1)-x(4))));
14          -x(7) + Pload4 + G44*V_pv4(i+1)*V_pv4(i+1) +
               V_pv4(i+1)*x(3)*(G42*(cosd(angle_pv4(i+1)-x
               (4))) + B42*(sind(angle_pv4(i+1)-x(4))));
15          -x(8) + Qload4 - B44*V_pv4(i+1)*V_pv4(i+1) +
               V_pv4(i+1)*x(3)*(G42*(sind(angle_pv4(i+1)-x
               (4))) - B42*(cosd(angle_pv4(i+1)-x(4))));
16          -x(9) + Pload5 + G55*V_pv5(i+1)*V_pv5(i+1) +
               V_pv5(i+1)*x(3)*(G52*(cosd(angle_pv5(i+1)-x
               (4))) + B52*(sind(angle_pv5(i+1)-x(4))));
17          -x(10) + Qload5 - B55*V_pv5(i+1)*V_pv5(i+1) +
               V_pv5(i+1)*x(3)*(G52*(sind(angle_pv5(i+1)-x
               (4))) - B52*(cosd(angle_pv5(i+1)-x(4))));
18          G22*x(3)*x(3) + x(3)*x(1)*(G21*(cosd(x(4)-x(2))
               ) + B21*(sind(x(4)-x(2)))) ...
19                        + x(3)*V_pv3(i+1)*(G23*(cosd(x(4)
                             -angle_pv3(i+1))) + B23*(sind(
                             x(4)-angle_pv3(i+1)))) ...
20                        + x(3)*V_pv4(i+1)*(G24*(cosd(x(4)
                             -angle_pv4(i+1))) + B24*(sind(
                             x(4)-angle_pv4(i+1)))) ...
21                        + x(3)*V_pv5(i+1)*(G25*(cosd(x(4)
                             -angle_pv5(i+1))) + B25*(sind(
                             x(4)-angle_pv5(i+1)))); ...
22          -B22*x(3)*x(3) + x(3)*x(1)*(G21*(sind(x(4)-x(2)
               )) - B21*(cosd(x(4)-x(2)))) ...
23                         + x(3)*V_pv3(i+1)*(G23*(sind(x
                              (4)-angle_pv3(i+1))) - B23*(
                              cosd(x(4)-angle_pv3(i+1))))
                              ...
```

```
24                                  + x(3)*V_pv4(i+1)*(G24*(sind(x
                                        (4)-angle_pv4(i+1))) - B24*(
                                        cosd(x(4)-angle_pv4(i+1))))
                                        ...
25                                  + x(3)*V_pv5(i+1)*(G25*(sind(x
                                        (4)-angle_pv5(i+1))) - B25*(
                                        cosd(x(4)-angle_pv5(i+1))))
                                        ...
26          ];
27  end
```

## B.5 Load Function

```
1  %load function
2
3  datapointsX = secs/h;
4
5  data=0:datapointsX;
6
7  P1_hrs = hrs/6;
8  P1_datapoints = ((datapointsX)*(P1_hrs/hrs))+1;
9  P1 = 0.5 + (1-exp(0.000955*h*data(1:P1_datapoints)))*1e
       -4; %7am to 9am
10 Q1 = 0.1 + (1-exp(0.000955*h*0.9*data(1:P1_datapoints))
       )*1e-4; %7am to 9am
11
12 P2_hrs = hrs/3;
13 P2_datapoints = ((datapointsX)*(P2_hrs/hrs));
14 end2_datapoint = P1_datapoints+P2_datapoints;
15 P2 = P1(end)+(exp(-1e-3*h*(data(P1_datapoints:
       end2_datapoint)-data(P1_datapoints)))-1)*1e-4;%9am
       to 1pm
16 Q2 = Q1(end)+(exp(-1e-3*h*0.9*(data(P1_datapoints:
       end2_datapoint)-data(P1_datapoints)))-1)*1e-4;%9am
       to 1pm
```

```
17
18 P3_hrs = hrs/3;
19 P3_datapoints = ((datapointsX)*(P3_hrs/hrs));
20 end3_datapoint = end2_datapoint+P3_datapoints;
21 P3 = P2(end)+(exp(0.001115*(h/2)*(data(end2_datapoint:
       end3_datapoint)-data(end2_datapoint)))-1)*1e-4;%1pm
       to 5pm
22 Q3 = Q2(end)+(exp(0.001115*(h/2)*0.9*(data(
       end2_datapoint:end3_datapoint)-data(end2_datapoint))
       )-1)*1e-4;%1pm to 5pm
23
24 P4_hrs = hrs/6;
25 P4_datapoints = ((datapointsX)*(P4_hrs/hrs));
26 end4_datapoint = end3_datapoint+P4_datapoints;
27 P4 = P3(end)+ (1-exp(-0.000955*h*(data(end3_datapoint:
       end4_datapoint)-data(end3_datapoint))))*1e-4;%5pm to
        7pm
28 Q4 = Q3(end)+ (1-exp(-0.000955*h*0.9*(data(
       end3_datapoint:end4_datapoint)-data(end3_datapoint))
       ))*1e-4;%5pm to 7pm
29
30 PloadX(1:P1_datapoints) = P1;
31 PloadX(P1_datapoints:end2_datapoint) = P2;
32 PloadX(end2_datapoint:end3_datapoint) = P3;
33 PloadX(end3_datapoint:end4_datapoint) = P4;
34 QloadX(1:P1_datapoints) = Q1;
35 QloadX(P1_datapoints:end2_datapoint) = Q2;
36 QloadX(end2_datapoint:end3_datapoint) = Q3;
37 QloadX(end3_datapoint:end4_datapoint) = Q4;
38
39 % figure(3)
40 % subplot(2,1,1)
41 % plot(PloadX)
42 % subplot(2,1,2)
43 % plot(QloadX)
44
```

```
% subplot(4,1,1)
% plot(P1)
% subplot(4,1,2)
% plot(P2)
% subplot(4,1,3)
% plot(P3)
% subplot(4,1,4)
% plot(P4)
% figure(2)
% subplot(4,1,1)
% plot(Q1)
% subplot(4,1,2)
% plot(Q2)
% subplot(4,1,3)
% plot(Q3)
% subplot(4,1,4)
% plot(Q4)
```

## B.6 Central controller

```
global  i


%AGC

%Observe system frequency increase from ws (assuming
    wpv = wg and droop is in steady state)
%Delta_omega = wg-ws = (-delta_Pg - delta_Ppv)/( (1/(RD
    *ws))((Sgen_RATED/Spv_RATED)+MPP(i+1)) )

%Reset BattMPP and Qrated
BattMPP3(i+1) = MPP3(i+1);
Qrated3(i+1) = sqrt(1^2 - BattMPP3(i+1)^2);
BattMPPp3 = ( (X23pu*BattMPP3(i+1))-(R23pu*Qrated3(i+1)
    ) )/abs(Z23pu);
```

```
13 Qratedp3 = ( (R23pu*BattMPP3(i+1))+(X23pu*Qrated3(i+1))
       )/abs(Z23pu);
14
15 BattMPP4(i+1) = MPP4(i+1);
16 Qrated4(i+1) = sqrt(1^2 - BattMPP4(i+1)^2);
17 BattMPPp4 = ( (X24pu*BattMPP4(i+1))-(R24pu*Qrated4(i+1)
     ) )/abs(Z24pu);
18 Qratedp4 = ( (R24pu*BattMPP4(i+1))+(X24pu*Qrated4(i+1))
       )/abs(Z24pu);
19
20 BattMPP5(i+1) = MPP5(i+1);
21 Qrated5(i+1) = sqrt(1^2 - BattMPP5(i+1)^2);
22 BattMPPp5 = ( (X25pu*BattMPP5(i+1))-(R25pu*Qrated5(i+1)
     ) )/abs(Z25pu);
23 Qratedp5 = ( (R25pu*BattMPP5(i+1))+(X25pu*Qrated5(i+1))
       )/abs(Z25pu);
24
25 %Measured Variables
26 %MPP(i+1), P_pv3, P_pv4, P_pv5
27 DELTA_w = w_sys(i+1)-ws;
28
29 Pline = Pg(i+1);
30
31 DELTA_Vpv3 = V_pv3(i+1)-1;
32 DELTA_Vpv4 = V_pv4(i+1)-1;
33 DELTA_Vpv5 = V_pv5(i+1)-1;
34
35 %What do we need to increase each area generation set
     point by?
36 DELTA_Ppvp_SetPoint = -DELTA_w/(RD*ws);
                                                %(P_pv/BattMPP -
     PC_pv)
37 DELTA_PSV_SetPoint = -DELTA_w/(RD*ws);
38 DELTA_PC_pvQ = -DELTA_Vpv3/RD_Q;
                                                %Vpv = 1
    - RD_Q*((Qpvp/Qratedp) - PC_Q )
```

```
39 % DELTA_PC_Q4 = -DELTA_Vpv4/RD_Q;                                    %Vpv = 1 -
       RD_Q*((Qpvp/Qratedp) - PC_Q )
40 % DELTA_PC_Q5 = -DELTA_Vpv5/RD_Q;                                    %Vpv = 1 -
       RD_Q*((Qpvp/Qratedp) - PC_Q )
41
42 if Turn_off_SG(i)==0
43     P3 = BattMPP3(i+1)*(PC_pv3 + DELTA_Ppvp_SetPoint);                %New PV system setpoint(
       unnormalized)
44     P4 = BattMPP4(i+1)*(PC_pv4 + DELTA_Ppvp_SetPoint);                %New PV system setpoint(
       unnormalized)
45     P5 = BattMPP5(i+1)*(PC_pv5 + DELTA_Ppvp_SetPoint);                %New PV system setpoint(
       unnormalized)
46 elseif Turn_off_SG(i)==1
47     DELTA_Ppvp_SetPoint = -DELTA_w/(RD_pv*ws);                        %(change in P_pvp/
       MPPp)
48     Qp = PC_pvQ + DELTA_PC_pvQ;
49
50     Pp3 = BattMPPp3*(PC_pv3 + DELTA_Ppvp_SetPoint);
51     Qp3 = Qratedp3*Qp;
52     P3 = INVAmat23(1,:)*[Pp3;Qp3];
53
54     Pp4 = BattMPPp4*(PC_pv4 + DELTA_Ppvp_SetPoint);
55     Qp4 = Qratedp4*Qp;
56     P4 = INVAmat24(1,:)*[Pp4;Qp4];
57
58     Pp5 = BattMPPp5*(PC_pv5 + DELTA_Ppvp_SetPoint);
59     Qp5 = Qratedp5*Qp;
60     P5 = INVAmat25(1,:)*[Pp5;Qp5];
61 end
62
```

```
63 %How can we maximize power from PhotoVoltaic?
64 DELTA_Ppv_max(i+1) = ( MPP3(i+1)+MPP4(i+1)+MPP5(i+1) )
       - ( P3+P4+P5 );       %difference between MPP and
       Power generated (same as MPP-P_pv)
65 %Compare the maximum power output increase available to
       the required and
66 %make decisions based on that
67 if Pline>=DELTA_Ppv_max(i+1)
68     DELTA_P = DELTA_Ppv_max(i+1);
69 elseif Pline<DELTA_Ppv_max(i+1)
70     DELTA_P = Pline;
71 end
72
73 ED_total = MPP3(i+1)+MPP4(i+1)+MPP5(i+1);
74 ED_3 = MPP3(i+1)/ED_total;
75 ED_4 = MPP4(i+1)/ED_total;
76 ED_5 = MPP5(i+1)/ED_total;
77
78 if rem(t,60)==0
79     PC_pvQ = PC_pvQ + DELTA_PC_pvQ;
80 end
81
82 if Turn_off_SG(i)==0
83
84
85     PC_pv3 = PC_pv3 + DELTA_Ppvp_SetPoint + (ED_3*
           DELTA_P/BattMPP3(i+1));    %(ACEpv=
           DELTA_Ppv_SetPoint + DELTA_P)
86     PC_pv4 = PC_pv4 + DELTA_Ppvp_SetPoint + (ED_4*
           DELTA_P/BattMPP4(i+1));    %(ACEpv=
           DELTA_Ppv_SetPoint + DELTA_P)
87     PC_pv5 = PC_pv5 + DELTA_Ppvp_SetPoint + (ED_5*
           DELTA_P/BattMPP5(i+1));    %(ACEpv=
           DELTA_Ppv_SetPoint + DELTA_P)
88
89
```

```
90        P_pvp3 = PC_pv3*BattMPP3(i+1);
91        P_pv3(i+1) = P_pvp3;
92
93        P_pvp4 = PC_pv4*BattMPP4(i+1);
94        P_pv4(i+1) = P_pvp4;
95
96        P_pvp5 = PC_pv5*BattMPP5(i+1);
97        P_pv5(i+1) = P_pvp5;
98
99
100       Qpv3 = ( Q_pvp3 - (Amat23(2,1)*P_pv3(i+1)) )/Amat23
              (2,2);
101       Qpv4 = ( Q_pvp4 - (Amat24(2,1)*P_pv4(i+1)) )/Amat24
              (2,2);
102       Qpv5 = ( Q_pvp3 - (Amat25(2,1)*P_pv5(i+1)) )/Amat25
              (2,2);
103
104
105       w_pv3(i+1) = ws - m_pv3(i+1)*(-PC_pv3*BattMPP3(i+1)
              + P_pv3(i+1));
106       w_pv4(i+1) = ws - m_pv4(i+1)*(-PC_pv4*BattMPP4(i+1)
              + P_pv4(i+1));
107       w_pv5(i+1) = ws - m_pv5(i+1)*(-PC_pv5*BattMPP5(i+1)
              + P_pv5(i+1));
108
109   end
110
111   PC = PC + DELTA_PSV_SetPoint - (DELTA_P*NET_BASE/
          Sgen_RATED);          %(ACEpv=DELTA_PSV_SetPoint -
          DELTA_P)
112
113   %PhotoVoltaic frequency control should kick in when
          Pline < DELTA_Ppv_max
114   if ( (MPP3(i+1) + MPP4(i+1) + MPP5(i+1))>(Pmax*3) )
115       Turn_off_SG(i+1) = 1;        %Turn off SG and see
              how system response changes
```

```
116         Pg(i+1) = 0;
117         Qg(i+1) = 0;
118         if Turn_off_SG(i)==0 && Turn_off_SG(i+1)==1      %SG
                was initially ON
119             PC_pv3 = ( PC_pv3*(BattMPP3(i+1) )/BattMPPp3);
120             PC_pv4 = ( PC_pv4*(BattMPP4(i+1) )/BattMPPp4);
121             PC_pv5 = ( PC_pv5*(BattMPP5(i+1) )/BattMPPp5);
                                        %renormalize setpoint
122         end
123
124         PC_pv3 = PC_pv3 + DELTA_Ppvp_SetPoint;    %(ACEpv=
              DELTA_Ppv_SetPoint + DELTA_P)
125         PC_pv4 = PC_pv4 + DELTA_Ppvp_SetPoint;    %(ACEpv=
              DELTA_Ppv_SetPoint + DELTA_P)
126         PC_pv5 = PC_pv5 + DELTA_Ppvp_SetPoint;    %(ACEpv=
              DELTA_Ppv_SetPoint + DELTA_P)
127 end
```

## B.7 Cloud Cover and Shading Model

```
1  % generate probability transitions
2
3  %Trans1
4  p1 = 400*ones(1,16) + 50*rand(1,16);
5
6  Trans1 = [p1(1)  60+5*rand  40+5*rand  27+5*rand
       20+10*rand  12+5*rand  8+4*rand  4.5+rand...
7              2.7+0.6*rand  2.4+0.3*rand  2.4+0.15*rand
                  1.8+0.4*rand  1.8+0.3*rand  1.8+0.2*rand
                  1.8+0.1*rand  1.8+0.1*rand ];
8  Trans1 = Trans1/sum(Trans1);
9
10
11 Trans2 = [50+5*rand  p1(2)  50+5*rand  35+5*rand  27+5*
       rand  20+10*rand  12+5*rand  8+4*rand...
```

```
12              4.5+rand   2.7+0.6*rand   2.4+0.3*rand
                    2.4+0.15*rand   1.8+0.4*rand   1.8+0.3*rand
                     1.8+0.2*rand   1.8+0.1*rand ];
13  Trans2 = Trans2/sum(Trans2);
14
15
16  Trans3 = [35+5*rand   50+5*rand   p1(3)   50+5*rand   35+5*
        rand   25+5*rand   20+10*rand   11+5*rand...
17              8+4*rand   4.5+rand   2.7+0.6*rand   2.4+0.3*
                    rand   2.4+0.15*rand   1.8+0.4*rand
                    1.8+0.3*rand   1.8+0.2*rand ];
18  Trans3 = Trans3/sum(Trans3);
19
20
21  Trans4 = [25+5*rand   35+5*rand   50+5*rand   p1(4)   50+5*
        rand   35+5*rand   25+5*rand   20+10*rand...
22              11+5*rand   8+4*rand   4.5+rand   2.7+0.6*rand
                    2.4+0.3*rand   2.4+0.15*rand   1.8+0.4*rand
                     1.8+0.3*rand ];
23  Trans4 = Trans4/sum(Trans4);
24
25
26  Trans5 = [20+10*rand   25+5*rand   35+5*rand   50+5*rand
        p1(5)   50+5*rand   35+5*rand   25+5*rand...
27              20+10*rand   11+5*rand   8+4*rand   4.5+rand
                    2.7+0.6*rand   2.4+0.3*rand   2.4+0.15*rand
                     1.8+0.4*rand ];
28  Trans5 = Trans5/sum(Trans5);
29
30
31  Trans6 = [11+5*rand   20+10*rand   25+5*rand   35+5*rand
        50+5*rand   p1(6)   50+5*rand   35+5*rand...
32              25+5*rand   20+10*rand   11+5*rand   8+4*rand
                    4.5+rand   2.7+0.6*rand   2.4+0.3*rand
                    2.4+0.15*rand ];
33  Trans6 = Trans6/sum(Trans6);
```

```
34
35
36 Trans7 = [8+4*rand  11+5*rand  20+10*rand  25+5*rand
      35+5*rand  50+5*rand  p1(7)  50+5*rand...
37            35+5*rand  25+5*rand  20+10*rand  11+5*rand
                 8+4*rand  4.5+rand  2.7+0.6*rand  2.4+0.3*
                 rand ];
38 Trans7 = Trans7/sum(Trans7);
39
40
41 Trans8 = [4.5+rand  8+4*rand  11+5*rand  20+10*rand
      25+5*rand  35+5*rand  50+5*rand  p1(8)...
42            50+5*rand  35+5*rand  25+5*rand  20+10*rand
                 11+5*rand  8+4*rand  4.5+rand  2.7+0.6*
                 rand ];
43 Trans8 = Trans8/sum(Trans8);
44
45
46 Trans9 = [2.7+0.6*rand  4.5+rand  8+4*rand  11+5*rand
      20+10*rand  25+5*rand  35+5*rand  50+5*rand...
47            p1(9)  50+5*rand  35+5*rand  25+5*rand
                 20+10*rand  11+5*rand  8+4*rand  4.5+rand
                 ];
48 Trans9 = Trans9/sum(Trans9);
49
50
51 Trans10 = [2.4+0.3*rand  2.7+0.6*rand  4.5+rand  8+4*
      rand  11+5*rand  20+10*rand  25+5*rand  35+5*rand...
52              50+5*rand  p1(10)  50+5*rand  35+5*rand
                   25+5*rand  20+10*rand  11+5*rand  8+4*
                   rand ];
53 Trans10 = Trans10/sum(Trans10);
54
55
56 Trans11 = [1.8+0.4*rand  2.4+0.3*rand  2.7+0.6*rand
      4.5+rand  8+4*rand  11+5*rand  20+10*rand  25+5*rand
```

```
    ...
57              35+5*rand   50+5*rand   p1(11)   50+5*rand
                    35+5*rand   25+5*rand   20+10*rand   11+5*
                    rand ];
58  Trans11 = Trans11/sum(Trans11);
59
60
61  Trans12 = [1.8+0.3*rand   1.8+0.4*rand   2.4+0.3*rand
        2.7+0.6*rand   4.5+rand   8+4*rand   11+5*rand   20+10*
        rand...
62              25+5*rand   35+5*rand   50+5*rand   p1(12)
                    50+5*rand   35+5*rand   25+5*rand   20+10*
                    rand];
63  Trans12 = Trans12/sum(Trans12);
64
65
66  Trans13 = [1.8+0.2*rand   1.8+0.3*rand   1.8+0.4*rand
        2.4+0.3*rand   2.7+0.6*rand   4.5+rand   8+4*rand
        11+5*rand...
67              20+10*rand   25+5*rand   35+5*rand   50+5*
                    rand   p1(13)   50+5*rand   35+5*rand   25+5*
                    rand];
68  Trans13 = Trans13/sum(Trans13);
69
70
71  Trans14 = [1.8+0.1*rand   1.8+0.2*rand   1.8+0.3*rand
        1.8+0.4*rand   2.4+0.3*rand   2.7+0.6*rand   4.5+rand
        8+4*rand...
72              11+5*rand   20+10*rand   25+5*rand   35+5*
                    rand   50+5*rand   p1(14)   50+5*rand
                    35+5*rand];
73  Trans14 = Trans14/sum(Trans14);
74
75
76  Trans15 = [1.8+0.1*rand   1.8+0.2*rand   1.8+0.3*rand
        1.8+0.4*rand   2.4+0.15*rand   2.4+0.3*rand   2.7+0.6*
```

```
    rand   4.5+rand...
77              8+4*rand   11+5*rand    20+10*rand    25+5*rand
                    35+5*rand    50+5*rand   p1(15)   50+5*
                rand];
78  Trans15 = Trans15/sum(Trans15);
79
80
81  Trans16 = [1.8+0.1*rand   1.8+0.1*rand   1.8+0.2*rand
    1.8+0.3*rand   1.8+0.4*rand   2.4+0.15*rand   2.4+0.3*
    rand   2.7+0.6*rand...
82              4.5+rand   8+4*rand   12+5*rand    20+10*rand
                    27+5*rand   40+5*rand    60+5*rand   p1
                (16)];
83  Trans16 = Trans16/sum(Trans16);
84
85
86  %
87  p2 = 200*ones(1,16) + 50*rand(1,16);
88
89  TransSecond1 = [p2(1)   60+5*rand   40+5*rand   27+5*rand
        25+10*rand   12+5*rand   8+4*rand   4.5+rand...
90              2.7+0.6*rand   2.4+0.3*rand   2.4+0.15*rand
                1.8+0.4*rand   1.8+0.3*rand   1.8+0.2*rand
                1.8+0.1*rand   1.8+0.1*rand ];
91  TransSecond1 = TransSecond1/sum(TransSecond1);
92
93
94  TransSecond2 = [50+5*rand   p2(2)   50+5*rand   40+5*rand
        27+5*rand   25+10*rand   12+5*rand   8+4*rand...
95              4.5+rand   2.7+0.6*rand   2.4+0.3*rand
                2.4+0.15*rand   1.8+0.4*rand   1.8+0.3*rand
                1.8+0.2*rand   1.8+0.1*rand ];
96  TransSecond2 = TransSecond2/sum(TransSecond2);
97
98
```

```
99  TransSecond3 = [35+5*rand  50+5*rand  p2(3)  50+5*rand
        35+5*rand  27+5*rand  25+10*rand  12+5*rand...
100             8+4*rand  4.5+rand  2.7+0.6*rand  2.4+0.3*
                    rand  2.4+0.15*rand  1.8+0.4*rand
                    1.8+0.3*rand  1.8+0.2*rand ];
101 TransSecond3 = TransSecond3/sum(TransSecond3);
102
103
104 TransSecond4 = [27+5*rand  35+5*rand  50+5*rand  p2(4)
        50+5*rand  35+5*rand  27+5*rand  25+10*rand...
105             12+5*rand  8+4*rand  4.5+rand  2.7+0.6*rand
                    2.4+0.3*rand  2.4+0.15*rand  1.8+0.4*rand
                     1.8+0.3*rand ];
106 TransSecond4 = TransSecond4/sum(TransSecond4);
107
108
109 TransSecond5 = [20+10*rand  25+5*rand  35+5*rand  50+5*
      rand  p2(5)  50+5*rand  35+5*rand  25+5*rand...
110             20+10*rand  12+5*rand  8+4*rand  4.5+rand
                    2.7+0.6*rand  2.4+0.3*rand  2.4+0.15*rand
                     1.8+0.4*rand ];
111 TransSecond5 = TransSecond5/sum(TransSecond5);
112
113
114 TransSecond6 = [11+5*rand  20+10*rand  25+5*rand  35+5*
      rand  50+5*rand  p2(6)  50+5*rand  35+5*rand...
115             25+5*rand  20+10*rand  11+5*rand  8+4*rand
                    4.5+rand  2.7+0.6*rand  2.4+0.3*rand
                    2.4+0.15*rand ];
116 TransSecond6 = TransSecond6/sum(TransSecond6);
117
118
119 TransSecond7 = [8+4*rand  11+5*rand  20+10*rand  25+5*
      rand  35+5*rand  50+5*rand  p2(7)  50+5*rand...
120             35+5*rand  25+5*rand  20+10*rand  11+5*rand
                    8+4*rand  4.5+rand  2.7+0.6*rand  2.4+0.3*
```

```
                rand ];
121 TransSecond7 = TransSecond7/sum(TransSecond7);
122
123
124 TransSecond8 = [4.5+rand 8+4*rand 11+5*rand 20+10*
    rand 25+5*rand 35+5*rand 50+5*rand p2(8)...
125             50+5*rand 35+5*rand 25+5*rand 20+10*rand
                11+5*rand 8+4*rand 4.5+rand 2.7+0.6*
                rand ];
126 TransSecond8 = TransSecond8/sum(TransSecond8);
127
128
129 TransSecond9 = [2.7+0.6*rand 4.5+rand 8+4*rand 11+5*
    rand 20+10*rand 25+5*rand 35+5*rand 50+5*rand...
130             p2(9) 50+5*rand 35+5*rand 25+5*rand
                20+10*rand 11+5*rand 8+4*rand 4.5+rand
                ];
131 TransSecond9 = TransSecond9/sum(TransSecond9);
132
133
134 TransSecond10 = [2.4+0.3*rand 2.7+0.6*rand 4.5+rand
    8+4*rand 11+5*rand 20+10*rand 25+5*rand 35+5*
    rand...
135             50+5*rand p2(10) 50+5*rand 35+5*rand
                25+5*rand 20+10*rand 11+5*rand 8+4*
                rand ];
136 TransSecond10 = TransSecond10/sum(TransSecond10);
137
138
139 TransSecond11 = [1.8+0.4*rand 2.4+0.3*rand 2.7+0.6*
    rand 4.5+rand 8+4*rand 11+5*rand 20+10*rand
    25+5*rand...
140             35+5*rand 50+5*rand p2(11) 50+5*rand
                35+5*rand 25+5*rand 20+10*rand 11+5*
                rand ];
141 TransSecond11 = TransSecond11/sum(TransSecond11);
```

```
142
143
144 TransSecond12 = [1.8+0.3*rand  1.8+0.4*rand  2.4+0.3*
    rand  2.7+0.6*rand  4.5+rand  8+4*rand  11+5*rand
    20+10*rand...
145             25+5*rand  35+5*rand  50+5*rand  p2(12)
                50+5*rand  35+5*rand  25+5*rand  20+10*
                rand];
146 TransSecond12 = TransSecond12/sum(TransSecond12);
147
148
149 TransSecond13 = [1.8+0.2*rand  1.8+0.3*rand  1.8+0.4*
    rand  2.4+0.3*rand  2.7+0.6*rand  4.5+rand  8+4*rand
      11+5*rand...
150             25+10*rand  25+5*rand  35+5*rand  50+5*
                rand  p2(13)  50+5*rand  35+5*rand  25+5*
                rand];
151 TransSecond13 = TransSecond13/sum(TransSecond13);
152
153
154 TransSecond14 = [1.8+0.1*rand  1.8+0.2*rand  1.8+0.3*
    rand  1.8+0.4*rand  2.4+0.3*rand  2.7+0.6*rand  4.5+
    rand  8+4*rand...
155             12+5*rand  25+10*rand  27+5*rand  35+5*
                rand  50+5*rand  p2(14)  50+5*rand
                35+5*rand];
156 TransSecond14 = TransSecond14/sum(TransSecond14);
157
158
159 TransSecond15 = [1.8+0.1*rand  1.8+0.2*rand  1.8+0.3*
    rand  1.8+0.4*rand  2.4+0.15*rand  2.4+0.3*rand
    2.7+0.6*rand  4.5+rand...
160             8+4*rand  12+5*rand  25+10*rand  27+5*rand
                  35+5*rand  50+5*rand  p2(15)  50+5*
                rand];
161 TransSecond15 = TransSecond15/sum(TransSecond15);
```

```
162
163
164 TransSecond16 = [1.8+0.1*rand  1.8+0.1*rand  1.8+0.2*
      rand  1.8+0.3*rand  1.8+0.4*rand  2.4+0.15*rand
      2.4+0.3*rand  2.7+0.6*rand...
165               4.5+rand  8+4*rand  12+5*rand  25+10*rand
                      27+5*rand  40+5*rand  60+5*rand  p2
                    (16)];
166 TransSecond16 = TransSecond16/sum(TransSecond16);
167
168
169 %
170
171 p3 = 200*ones(1,5) + 50*rand(1,5);
172
173 TransSlow1 = [p3(1)  50+5*rand  12.5+5*rand  3.125+5*
      rand  0.78+5*rand];
174 TransSlow1 = TransSlow1/sum(TransSlow1);
175
176
177 TransSlow2 = [25+5*rand  p3(2)  25+5*rand  13.28+5*rand
       3.125+5*rand];
178 TransSlow2 = TransSlow2/sum(TransSlow2);
179
180
181 TransSlow3 = [6.25+5*rand  25+5*rand  p3(3)  25+5*rand
      6.25+5*rand];
182 TransSlow3 = TransSlow3/sum(TransSlow3);
183
184
185 TransSlow4 = [3.125+5*rand  13.28+5*rand  25+5*rand  p3
      (4)  25+5*rand];
186 TransSlow4 = TransSlow4/sum(TransSlow4);
187
188
```

```
189 TransSlow5 = [0.78+5*rand   3.125+5*rand   12.5+5*rand
       25+5*rand   p3(5)];
190 TransSlow5 = TransSlow5/sum(TransSlow5);

  1 TransMatrix1 = [...
  2
  3 Trans1;...
  4 Trans2;...
  5 Trans3;...
  6 Trans4;...
  7 Trans5;...
  8 Trans6;...
  9 Trans7;...
 10 Trans8;...
 11 Trans9;...
 12 Trans10;...
 13 Trans11;...
 14 Trans12;...
 15 Trans13;...
 16 Trans14;...
 17 Trans15;...
 18 Trans16];
 19
 20 TransMatrix2 = [...
 21
 22 TransSecond1;...
 23 TransSecond2;...
 24 TransSecond3;...
 25 TransSecond4;...
 26 TransSecond5;...
 27 TransSecond6;...
 28 TransSecond7;...
 29 TransSecond8;...
 30 TransSecond9;...
 31 TransSecond10;...
 32 TransSecond11;...
```

```
TransSecond12;...
TransSecond13;...
TransSecond14;...
TransSecond15;...
TransSecond16];

TransMatrix3 = [TransSlow1; TransSlow2; TransSlow3;
    TransSlow4; TransSlow5];

%Cloud cover
%[0 5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65%
    70% 75%]

CloudCoverTransitionProbabilities

TransitionMatrices

%
%Determine pmf at each time instant
%

p = zeros(m,16);
p(1,:) = [1 zeros(1,15)];


for i = 2: 1 + ( (m-1)/3 )

    p(i,:) = p(i-1,:) * TransMatrix1;

end


for i = 2 + ( (m-1)/3 ): 1 + ( 2*(m-1)/3 )

    p(i,:) = p(i-1,:) * TransMatrix2;

```

```
27 end
28
29
30 for i = 2 + ( 2*(m-1)/3 ): m
31
32     p(i,:) = p(i-1,:) * TransMatrix1;
33
34 end
35
36 %        Determine Percentage Cloud Cover        %
37 Xx = rand(1,m);
38 Yy = zeros(1,16);
39 CloudCoverState = zeros(1,m);
40
41 for i = 1:m
42     for j = 1:16
43
44         Yy(j) = sum(p(i,1:j));
45
46 %          if rand<Yy(j)
47 %              CloudCoverState(i) = j;
48 %              break
49 %          end
50
51         if Xx(i)<Yy(j)
52             CloudCoverState(i) = j;
53             break
54         end
55
56     end
57 end
58
59 for i=1:m
60
61     CloudCoverState(i) = -( CloudCoverState(i)-1 )
       *0.01;          %maximum cloud cover = 15percent
```

```
62
63 end
64
65
66
67
68 %SLOW CLOUD COVER
69
70
71
72 % Every 30mins (24 times in 12hrs), the slow cloud
       cover kicks in
73 %[0 17.5% 35% 52.5% 70%]
74
75 SlowCloudTimeInterval = 5;                    %in minutes
76 TotalMinutes = 60*12;                          %12hr
       period
77 SlowCloudSteps = TotalMinutes/SlowCloudTimeInterval;
78 pSLOW = zeros(SlowCloudSteps,5);
79 pSLOW(1,:) = [1 zeros(1,4)];
80
81 %        Determine Percentage Cloud Cover        %
82 Xx2 = rand(1,SlowCloudSteps+1);
83 Yy2 = zeros(1,5);
84 CloudCoverStateSLOW = zeros(1,SlowCloudSteps+1);
85
86
87
88
89 for i = 2:SlowCloudSteps+1
90
91     pSLOW(i,:) = pSLOW(i-1,:) * TransMatrix3;
92
93 end
94
95 for i = 1:SlowCloudSteps+1
```

```
96      for j = 1:5
97
98          Yy2(j) = sum(pSLOW(i,1:j));
99
100         if rand<=Yy2(j)
101             CloudCoverStateSLOW(i) = j;
102             break
103         end
104
105 %           if Xx2(i)<=Yy2(j)
106 %               CloudCoverStateSLOW(i) = j;
107 %               break
108 %           end
109
110     end
111 end
112
113 for i=1:SlowCloudSteps+1
114
115     CloudCoverStateSLOW(i) = -( CloudCoverStateSLOW(i)
        -1 )*(17.5/100);          %maximum cloud cover =
        70percent
116
117 end
118
119
120 SlowCloudInterval = (m-1)/SlowCloudSteps;
121 SlowPerTimeStep = 0.7/( (m-1)/144 );
```

# REFERENCES


[1] M. Villalva, J. Gazoli, and E. Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *Power Electronics, IEEE Transactions on*, vol. 24, no. 5, pp. 1198–1208, 2009.

[2] A. Yazdani and R. Iravani, *Voltage-Sourced Converters in Power Systems*. Wiley, 2010.

[3] T. Vandoorn, J. D. Kooning, B. Meersman, and L. Vandevelde, "Review of primary control strategies for islanded microgrids with power-electronic interfaces," *Renewable and Sustainable Energy Reviews*, vol. 19, no. 0, pp. 613 – 628, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032112006764

[4] I. Balaguer, H.-G. Kim, F. Peng, and E. Ortiz, "Survey of photovoltaic power systems islanding detection methods," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, 2008, pp. 2247–2252.

[5] P. Krein and T. Esram, "Admittance control for anti-islanding protection in modular grid-connected inverters," in *Control and Modeling for Power Electronics (COMPEL), 2010 IEEE 12th Workshop on*, 2010, pp. 1–5.

[6] Y. W. Li and C.-N. Kao, "An accurate power control strategy for power-electronics-interfaced distributed generation units operating in a low-voltage multibus microgrid," *Power Electronics, IEEE Transactions on*, vol. 24, no. 12, pp. 2977–2988, 2009.

[7] R. Lasseter, "Microgrids," in *Power Engineering Society Winter Meeting, 2002. IEEE*, vol. 1, 2002, pp. 305–308 vol.1.

[8] H. Xin, Y. Liu, Z. Wang, D. Gan, and T. Yang, "A new frequency regulation strategy for photovoltaic systems without energy storage," *Sustainable Energy, IEEE Transactions on*, vol. 4, no. 4, pp. 985–993, 2013.

[9] C. Hill, M. Such, D. Chen, J. Gonzalez, and W. Grady, "Battery energy storage for enabling integration of distributed solar power generation," *Smart Grid, IEEE Transactions on*, vol. 3, no. 2, pp. 850–857, 2012.

[10] N. Kakimoto, S. Takayama, H. Satoh, and K. Nakamura, "Power modulation of photovoltaic generator for frequency control of power system," *Energy Conversion, IEEE Transactions on*, vol. 24, no. 4, pp. 943–949, 2009.

[11] W. Omran, M. Kazerani, and M. Salama, "Investigation of methods for reduction of power fluctuations generated from large grid-connected photovoltaic systems," *Energy Conversion, IEEE Transactions on*, vol. 26, no. 1, pp. 318–327, 2011.

[12] M. Datta, T. Senjyu, A. Yona, T. Funabashi, and C.-H. Kim, "A frequency-control approach by photovoltaic generator in a pv-diesel hybrid power system," *Energy Conversion, IEEE Transactions on*, vol. 26, no. 2, pp. 559–571, 2011.

[13] Y. Liu, L. Chen, L. Chen, H. Xin, and D. Gan, "A newton quadratic interpolation based control strategy for photovoltaic system," in *Sustainable Power Generation and Supply (SUPERGEN 2012), International Conference on*, 2012, pp. 1–6.

[14] A. Yazdani, A. Di Fazio, H. Ghoddami, M. Russo, M. Kazerani, J. Jatskevich, K. Strunz, S. Leva, and J. Martinez, "Modeling guidelines and a benchmark for power system simulation studies of three-phase single-stage photovoltaic systems," *Power Delivery, IEEE Transactions on*, vol. 26, no. 2, pp. 1247–1264, 2011.

[15] K. Kim, P. Krein, J. Lee, H. Bae, and B.-H. Cho, "Irradiance and temperature transient sensitivity analysis for photovoltaic control," in *Power Electronics and ECCE Asia (ICPE ECCE), 2011 IEEE 8th International Conference on*, 2011, pp. 393–400.

[16] A. Yazdani, "Electromagnetic transients of grid-tied photovoltaic systems based on detailed and averaged models of the voltage-sourced converter," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1–8.

[17] P. Krein, *Elements Of Power Electronics*, ser. The Oxford Series in Electrical and Computer Engineering Series. Oxford University Press, Incorporated, 1998.

[18] H. Li, Y. Xu, S. Adhikari, D. Rizy, F. Li, and P. Irminger, "Real and reactive power control of a three-phase single-stage pv system and pv voltage stability," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1–8.

[19] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, ser. Alternative Etext Formats. Pearson, 2010.

[20] P. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery*, ser. IEEE Press Series on Power Engineering (was Power Systems Engineering), Series Editor: Paul M. Anderson. IEEE Press, 1995.

[21] N. Pogaku, M. Prodanovic, and T. Green, "Modeling, analysis and testing of autonomous operation of an inverter-based microgrid," *Power Electronics, IEEE Transactions on*, vol. 22, no. 2, pp. 613–625, 2007.

[22] K. De Brabandere, B. Bolsens, J. Van den Keybus, A. Woyte, J. Driesen, and R. Belmans, "A voltage and frequency droop control method for parallel inverters," *Power Electronics, IEEE Transactions on*, vol. 22, no. 4, pp. 1107–1115, 2007.

[23] A. Fitzgerald, A. Fitzgerald, C. Kingsley, and S. Umans, *Electric Machinery*, ser. Electrical Engineering Series. McGraw-Hill, 2003.

[24] S. Chapman, *Electric Machinery Fundamentals*. McGraw-Hill, 2012.

[25] C. Ong, *Dynamic Simulation of Electric Machinery: Using MATLAB/SIMULINK*. Prentice Hall PTR, 1998.

[26] P. Sauer and A. Pai, *Power System Dynamics and Stability*. Stipes Publishing L.L.C., 2006.

[27] J. Glover, M. Sarma, and T. Overbye, *Power System Analysis and Design*. Cengage Learning, 2011.

[28] E. Barklund, N. Pogaku, M. Prodanovic, C. Hernandez-Aramburo, and T. Green, "Energy management system with stability constraints for stand-alone autonomous microgrid," in *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*, 2007, pp. 1–6.

[29] Y.-R. Mohamed and E. El-Saadany, "Adaptive decentralized droop controller to preserve power sharing stability of paralleled inverters in distributed generation microgrids," *Power Electronics, IEEE Transactions on*, vol. 23, no. 6, pp. 2806–2816, 2008.

[30] S. Iyer, M. Belur, and M. Chandorkar, "A generalized computational method to determine stability of a multi-inverter microgrid," *Power Electronics, IEEE Transactions on*, vol. 25, no. 9, pp. 2420–2432, 2010.

[31] X. Sun, Y.-S. Lee, and D. Xu, "Modeling, analysis, and implementation of parallel multi-inverter systems with instantaneous average-current-sharing scheme," *Power Electronics, IEEE Transactions on*, vol. 18, no. 3, pp. 844–856, 2003.

[32] M. Chandorkar, D. Divan, and R. Adapa, "Control of parallel connected inverters in standalone ac supply systems," *Industry Applications, IEEE Transactions on*, vol. 29, no. 1, pp. 136–143, 1993.

[33] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. Wiley, 1984.

[34] Q. Zhong and T. Hornik, *Control of Power Inverters in Renewable Energy and Smart Grid Integration*. Wiley, 2013.